图和网络及其应用

费培之　编著



四川大学出版社

图和网络及其应用

费培之 编著

四川大学出版社

1996年·成都

(川) 新登字 014 号

责任编辑：石大明
封面设计：唐利民
技术设计：徐　明

内 容 简 介

本书共有九章，包括图的基本概念和基本理论及其应用，图在计算机中的表示和图的基本算法，图论模型，网络及其应用，网络计划技术，网络规划等.

书中给出了 60 个常用算法，612 道例题和习题，这对于提高学习兴趣，加深对内容的理解，增强独立思考和应用的能力是大有裨益的.

本书具有系统性、理论性、实用性和新颖性，结构安排合理，语言流畅清晰，可作工程与计算机科学、数学、应用数学、经济数学与运筹学等理工科有关专业本科生和研究生的教材，对于《数学模型》课程和全国大学生数学模型竞赛是一本非常适用的参考书，也可供有关科技人员阅读.

图和网络及其应用

费培之　编著
四川大学出版社出版发行（成都市望江路 29 号）
四川省新华书店经销　　郫县犀浦印刷厂印刷
787×1092mm 16 开本　17.25 印张　380 千字
1996 年 8 月第 1 版　　1996 年 8 月第 1 次印刷
印数：1—3000 册
ISBN 7-5614-1239-8/O·106　　定价 18.00 元

前 言

1736 年 Euler 解决七桥问题[19]标志图论的诞生. 经过整整二百年，图论的第一部专著——König 的《有限图与无限图的理论》[23]在 1936 年问世. 图论的迅速发展出现在近三十几年，这是与图论的广泛应用，以及计算机技术和电子技术的发展密切相关的.

图论是研究图的组合关系及结构的一个数学分支. 今天，图论在物理学、化学、计算机科学、通讯科学、经济学、运筹学、电气与土木工程、生物学、心理学、语言学、社会学与人类学等学科的应用已经是众所周知的. 事实上，凡是包含二元关系的系统都可以用一个图来模拟，或者说用图论来建立数学模型，并用图的理论和方法加以研究和解决. 由于图论在自然科学、社会科学、工程技术与经济管理等各方面的应用，并且图论具有形象直观的特点，所以受到人们的普遍重视，使图论在当今世界成为最活跃的数学分支之一.

本书是在作者多年从事本科生和研究生的教学中所使用的讲义的基础上，并参考近代有关文献编写而成的.

内容包括图和网络的基础、理论、方法、应用和算法，共有九章. 第一至五章是图的基本概念和基本理论及其应用. 第六章是图在计算机中的表示和图的基本算法. 第七章是图论模型，讨论如何把实际问题转化为图论问题来解决. 第八至九章是网络和网络规划及其应用.

图和网络的计算是在计算机上实现的. 本书在第六章讨论了图的基本算法，并在全书给 出了 60 个图和网络的常用算法.

各章都精心配置了例题和习题，全书共有 612 道. 这些题紧密配合正文，深浅适度，也有一些较难的题. 根据教学需要，可将部分习题提到正文中讲授.

为便于教学，在内容上自成体系，所需预备知识主要是线性代数，在第九章需要线性规划的有关知识，可参考文献 [12]，[13] 和 [14].

本书可作理工科有关专业本科生和研究生的教材. 注意到各校各专业对图和网络的要求不完全相同，在内容和结构的安排上尽可能方便使用者对教材的取舍. 可供选择的教学时数为一学期每周四学时左右，或者一学年每周三学时左右.

对于《数学模型》课程和全国大学生数学模型竞赛，这是一本非常适用的参考书. 也可供有关科技人员阅读.

本书是四川大学教材基金委员会重点资助的教材. 感谢校系领导和四川大学出版社的大力支持. 感谢石大明同志为本书出版付出的辛勤劳动.

由于水平有限，错漏之处不吝批评指正.

费 培 之
一九九五年七月于
四川大学数学系

符号说明

一般数学符号

符号	说明
R	实数集
R^n	n 维欧氏空间
$\varnothing$	空集
I	单位矩阵
$A_{i\cdot}$	矩阵 A 的 i 行
$A_{\cdot j}$	矩阵 A 的 j 列
$r(A)$	矩阵 A 的秩
$A>0$	矩阵 A 的每个元大于零
$A\geqslant 0$	矩阵 A 的每个元大于等于零
A^T	矩阵 A 的转置
$\det A$	方阵 A 的行列式
$f\|S$	函数 f 在 S 上的限制
$\|A\|$	集 A 的基数或元素个数
$\rho(A)$	集 A 的幂集
$a\in A$	a 是集 A 的元
$a\notin A$	a 不是集 A 的元
$\cup$	集的并
$\cap$	集的交
$\setminus$	集的差
$\oplus$	集的对称差
$\subseteq$	含于
$\subset$	含于且不等于
$[x]$	不大于 x 的最大整数
$\{x\}$	不小于 x 的是小整数
$\binom{n}{m}$	n 个选 m 个的组合数
$\gg$	远大于
$\langle\alpha,\beta\rangle$	向量 α 与 β 的内积
$\approx$	近似相等
$\triangleq$	定义为或表示为
$1\sim n$	从 1 到 n
n/m	n 除以 m
$\Longrightarrow$	必要的
$\Longleftarrow$	充分的
$\Longleftrightarrow$	充分必要的
$\vee$	逻辑加
$\wedge$	逻辑乘
u 和/或 v	u,或 v,或 u 与 v
$\dim V$	线性空间 V 的维数
$\bmod n$	模 n

图论符号

符号	说明
$V(G)$ 或 V	图 G 的顶点集
$E(G)$ 或 E	图 G 的边集
$\Gamma^+(v)$	顶点 v 的后继域
$\Gamma^-(v)$	顶点 v 的前趋域
$\Gamma(v)$	顶点 v 的邻域
$d_G^+(v)$ 或 $d^+(v)$	顶点 v 的出度
$d_G^-(v)$ 或 $d^-(v)$	顶点 v 的入度
$d_G(v)$ 或 $d(v)$	顶点 v 的度
$\delta(G)$ 或 δ	最小顶点度
$\delta^+(G)$ 或 δ^+	最小出度
$\delta^-(G)$ 或 δ^-	最小入度
$\Delta(G)$ 或 Δ	最大顶点度
$\Delta^+(G)$ 或 Δ^+	最大出度
$\Delta^-(G)$ 或 Δ^-	最大入度
$N^+(S)$	尾在 S 头不在 S 的边集
$N^-(S)$	头在 S 尾不在 S 的边集
$N(S)$ 或 $[S,\bar{S}]$	边割
(S,T)	尾在 S 头在 T 的边集
$[S,T]$	一个端点在 S 另一个端点在 T 的边集
K_n	n 个顶点的完全图
(X,Y,E)	偶图
$K_{r,s}$	完全偶图
$(V_1,\cdots,V_k,E)$	k 分图

$K_{r_1,\cdots,r_k}$	完全 k 分图
$G[V']$	点导出子图
$G[E']$	边导出子图
$G-V'$	删点子图
$G-E'$	删边子图
$G+E'$	加边图
$G\cdot e$	边 e 的缩并图
$H\cong G$	同构
$H\subseteq G$	子图
$H\subset G$	真子图
G^c	补图
$\overline{H}(G)$	H 在 G 的补图
$G_1\cup G_2$	子图的并
$G_1\cap G_2$	子图的交
$G-G_1$	子图的差
$G_1\oplus G_2$	子图的对称差
$H(G)$ 或 H	邻接矩阵
$A(G)$ 或 A	关联矩阵
A_f	基本关联矩阵
$B(G)$ 或 B	圈矩阵
B_f	基本圈矩阵
$Q(G)$ 或 Q	余圈矩阵
Q_f	基本余圈矩阵
$P(G)$ 或 P	可达矩阵
$D(G)$ 或 D	距离矩阵
$X(G)$ 或 X	控制矩阵
Φ	圈空间
Θ	余圈空间
$\Lambda(G)$	G 的自同构群
$b(G)$	块-分离点图
$c(v)$	顶点 v 的色度
val f	流 f 的值
cap K	割 K 的容量
$F(G)$	平面图 G 的面的集
$d(u,v)$	顶点 u 与 v 的距离
$d(G)$	图 G 的直径
$\sigma(G)$ 或 σ	图 G 的块数
$\varepsilon_0(G)$ 或 ε_0	图 G 的闭迹数
$\varepsilon_1(G)$ 或 ε_1	图 G 的圈数
$\nu_0(G)$ 或 ν_0	图 G 的边割数
$\nu_1(G)$ 或 ν_1	图 G 的余圈数
$\alpha_0(G)$ 或 α_0	图 G 的覆盖数
$\alpha_1(G)$ 或 α_1	图 G 的边覆盖数
$\beta_0(G)$ 或 β_0	图 G 的独立数
$\beta_1(G)$ 或 β_1	图 G 的匹配数
$\gamma_0(G)$ 或 γ_0	图 G 的控制数
$\omega(G)$ 或 ω	图 G 的连通支数
$\omega_0(G)$ 或 ω_0	图 G 的奇连通支数
$\kappa(G)$ 或 κ	图 G 的连通度
$\lambda(G)$ 或 λ	图 G 的边连通度
$\chi(G)$ 或 χ	图 G 的色数
$\chi'(G)$ 或 χ'	图 G 的边色数
$\xi(G)$ 或 ξ	图 G 的秩或余圈秩
$\eta(G)$ 或 η	图 G 的零度或圈秩
$\tau(G)$ 或 τ	图 G 的生成树数目
N	网络
$N(f)$	伴随网络
F	开关网络的传输矩阵
Z	开关网络的连接矩阵
W	开关网络的路矩阵
W_f	开关网络的基本路矩阵
$r(k,l)$	拉姆瑟数

目　　录

第七章　图论模型

第一章　图的基本概念

§1.1　什么是图

在 18 世纪的东普鲁士有一个城市叫哥尼斯堡，流经该城市有一条河，河中有两个小岛把市区分为四块陆地，陆地间有七座桥相连，如图 1.1 所示. 当时，人们在环城游览中总是在尝试并议论着一个伤脑筋的问题：能否从自己的居住地出发作环城游览，走过每座桥恰好一次最后回到出发地？所有的尝试都失败了，为什么？

瑞士数学家欧拉(Euler)在 1736 年发表了著名的论文"依据几何位置的解题方法"[19]，这是图论的第一篇论文，他标志图论的诞生. 欧拉用四个小圆圈代表四块陆地，用七条边代表七座桥，画出了七桥问题的模拟图，如图 1.2. 利用这个模拟图，欧拉指出：若每块陆地所连接的桥都是偶数座，则从任一陆地出发必能通过每座桥恰好一次而回到出发地，否则是不可能的. 由此，欧拉完全解决了七桥问题.

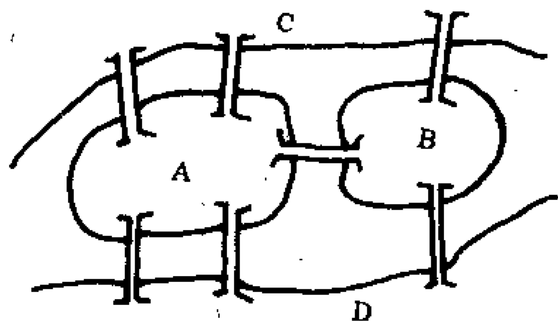


图 1.1　哥尼斯堡七桥示意图

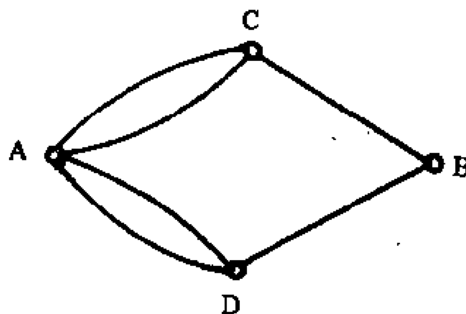


图 1.2　七桥问题模拟图

欧拉利用模拟图解决七桥问题的方法当时还是一种孤立现象，并未引起人们的普遍兴趣. 一个世纪之后，1847 年，克希霍夫在解决电网络方程中如何确定独立方程数目的问题时，提出了树的概念[20]. 如果把一棵树的树梢及树枝生长点作为图的"点"，树枝作为"边"，则在抽象意义下，树就是连通的、无圈的——此即是图论中的树的概念. 1857 年，凯莱在研究有机化学中碳水化合物的同分异构物时，也独立地发现了树[21]，如图 1.3. 对于树的研究是图的发展的一个重要标志.

图论中的另一个著名问题就是哈密顿圈，这个问题从 1858 年提出至今仍然没有完全解决. 1858 年哈密顿[22] 发明了一种被称为 Icosian 的游戏(Icosian 是个希腊字，意指 20). 游戏的方法是以十二面体的 20 个顶点代表世界上的 20 个城市，游戏者从某个城市出发，在十二面体上依次经过每个城市恰好一次，最后回到出发点. 十二面体如图 1.4，相应画在平面上的模拟图如图 1.5. 由此，这个游戏又称为环球旅行游戏.

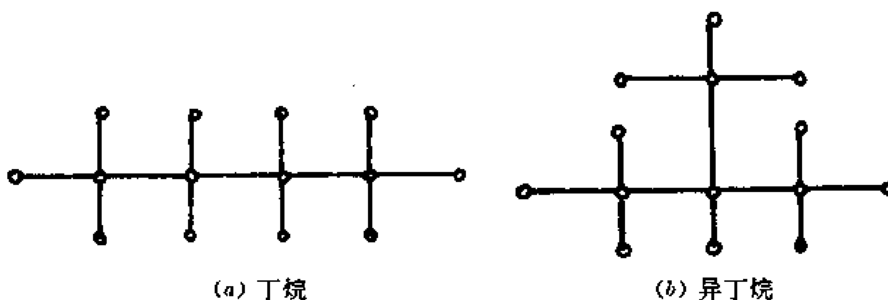


图 1.3 同分异构物的分子结构图

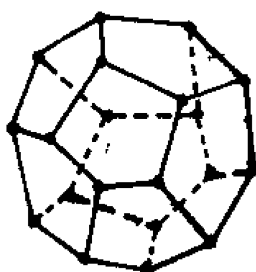


图 1.4 环球旅行游戏示意图

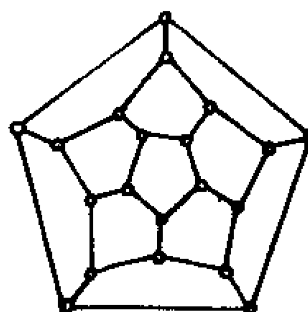


图 1.5 环球旅行游戏模拟图

对图的研究和应用虽早在 1736 年从欧拉研究七桥难题就开始了，但一直是点上的工作，没有面上的发展. 直到漫长的二百年过去之后的 1936 年，图论的第一本专著，匈牙利数学家 O. König 所著《有限图与无限图的理论》[23] 才得以问世.

图论的迅速发展是在本世纪五六十年代开始的. 大体说来这与电子技术和计算机技术的发展是同步的. 由此我们说图论是一门古老而年轻的学科，并且正显示其生机勃勃的活力，其发展及应用前景是十分明显的.

假定自然界和人类社会中的事物用一个小圆圈代表，称为顶点. 如果两事物间有某种关系，则用一条线段连接代表该事物的二个顶点，从而得到一个图，称为该系统的模拟图，正如七桥问题模拟图，环球旅行游戏模拟图等. 因此，一个具有二元关系的系统总可以用一个图来模拟. 直观地说，图就是由一些点（至少一个顶点）以及连接这些点的线段（直线段或曲线段）所组成的图形. 这些点称为**顶点**，这些线段称为**边**. 例如，我们常见的运输网络图，电网络图，通讯网络图，工程流程图，天燃气管网图，分子结构图，生物链图，人体经络图，神经网络图，管理系统图，组织机构图，销售连锁图，交通图（铁路、公路、航海、航空、航天等），导游图，家谱图等等. 总之，无论在自然科学，社会科学，工程技术与经济管理各方面，常可用图来描述事物及事物间的关系，并可根据对图的性质的分析来研究和揭示各种系统的特性，提供对系统进行决策的科学依据.

§1.2 图的定义

定义 1.1 称 $G=(V,E)$ 是一个图，如果

(1) V 是一个非空有限集合，①

(2) E 是 V 中元素的无序对所组成的有限集合，

并把 V 的元素叫做图的**顶点**，E 的元素叫做图的**边**.

图 $G=(V,E)$ 常简记为图 G，并以 $V(G)$ 或 V，$E(G)$ 或 E 分别表示图 G 的顶点集，边集. 设 S 是一个集合，则以 $|S|$ 表示集 S 的基数或元素个数. 于是 $|V(G)|$ 或 $|V|$，$|E(G)|$ 或 $|E|$ 分别表示图 G 的顶点数，边数. 由定义，边是顶点的无序对，若边 e 是顶点 u 和 v 的无序对，则记 $e=(u,v)$ 或 $e=(v,u)$.

设 $e=(u,v)$，则称 u,v 是 e 的**端点**，或 e 连接 u 与 v. 两个端点重合的边称为**环**. 若两条或两条以上的边有相同的端点，则这些边称为**平行边**. 若顶点 u 是边 e 的一个端点，则称顶点 u 与边 e **关联**. 不与任何边关联的顶点称为**孤立点**. 两个顶点称为**相邻**，如果它们与同一条边关联. 两条边称为**相邻**，如果它们与同一个顶点关联.

有 n 个顶点 m 条边的图称为 (n,m) 图. $(0,0)$ 图叫**空图**. $(n,0)$ 图称为**零图**，n 是正整数. 当 $n=1$，$(1,0)$ 图叫**平凡图**. 没有环与平行边的图称为**简单图**. 任意相异二顶点都相邻的简单图称为**完全图**. 有 n 个顶点的完全图记为 K_n. K_3 称为**三角形**. 设 $k\geqslant 2$ 为正整数，若 $V(G)$ 有一个分解 $V_1,V_2,\cdots,V_k$，即

$$V_1\cup V_2\cup\cdots\cup V_k=V(G),\quad V_i\cap V_j=\varnothing\quad 当\ i\neq j,$$

且每个 V_i 自身的顶点都互不相邻，则称图 G 为 **k 分图**，记为 $G=(V_1,\cdots,V_k,E)$；不在同一子集中的任意相异二顶点都相邻的简单 k 分图，称为**完全 k 分图**，记为 $K_{r_1,\cdots,r_k}$，其中 $|V_i|=r_i$，$i=1,\cdots,k$. 当 $k=2$ 时，2 分图又称**偶图**，完全 2 分图又称**完全偶图**.

定义 1.2 顶点 v 的**邻域** $\Gamma(v)$ 定义为

$$\Gamma(v)\triangleq\{u\in V(G)\,|\,u 与 v 相邻\}.$$

若 $S\subseteq V$，则 S 的**邻域** $\Gamma(S)$ 定义为

$$\Gamma(S)\triangleq\bigcup\{\Gamma(v)\,|\,v\in S\}.$$

显然 $\Gamma: V(G)\to\rho(V(G))$ 是一个映射，这里 $\rho(V(G))$ 表示 $V(G)$ 的幂集，即 $V(G)$ 的所有子集的集合.

对于简单图，边由顶点的相邻关系唯一决定，因此简单图 $G=(V,E)$ 可以表示为 $G=(V,\Gamma)$.

例 1.1 用映射 Γ 表示图 1.6 的简单图.

图 G 可表示为 $G=(V,\Gamma)$，这里

$\Gamma(v_1)=\{v_2,v_3\}$,

$\Gamma(v_2)=\{v_1,v_3,v_4\}$,

$\Gamma(v_3)=\{v_1,v_2\}$,

$\Gamma(v_4)=\{v_2\}$,

$\Gamma(v_5)=\varnothing$.

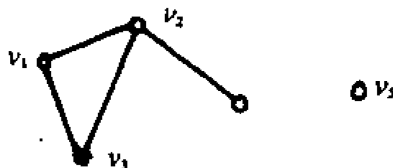


图 1.6 简单图 G

对任意的一个图 G，若去掉环，去掉平行边，则得到一个简单图，称为图 G 的**基图**.

---

① 为方便，当 $V=\varnothing$ 时称为空图.

例 1.2 求图 1.7(*a*) 的图的基图，如图 1.7(*b*).

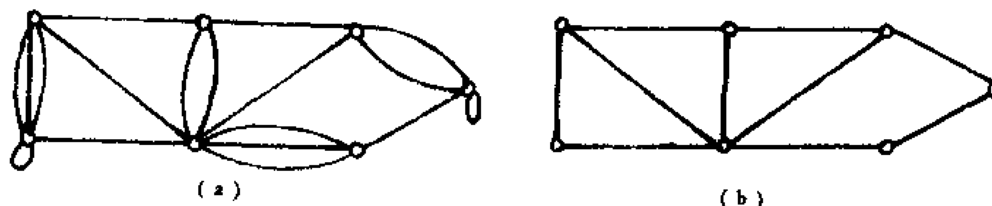


图 1.7 (*a*) 图 G (*b*) 图 G 的基图

§1.3 子图及其运算

定义 1.3 若 $V(H)\subseteq V(G)$，$E(H)\subseteq E(G)$，则称图 H 是图 G 的**子图**，G 是 H 的**母图**，记为 $H\subseteq G$. 若 $H\subseteq G$，$H\neq G$，则称 H 是 G 的**真子图**，记为 $H\subset G$. 若 $H\subseteq G$，$V(H)=V(G)$，则称 H 是 G 的**生成子图**.

设 $G=(V,E)$，若 $\varnothing\neq V'\subseteq V$，令

$$E'=\{(u,v)\in E\mid u,v\in V'\},$$

则子图 (V',E') 称为 G 的**由 V' 导出的子图**，简称为**点导出子图**或**导出子图**，记为 $G[V']$. 若 $V'=\{v_{i_1},\cdots,v_{i_k}\}$，则

$$[v_{i_1},\cdots,v_{i_k}]\triangleq G[V'].$$

若从 G 中删去 V' 以及所有与 V' 的顶点关联的边，所得到的子图称为 G 的**删点子图**，记为 $G-V'$. 若 $V'=\{v\}$，则

$$G-v\triangleq G-\{v\}.$$

设 $G=(V,E)$，若 $\varnothing\neq E'\subseteq E$，令

$$V'=\{v\in V\mid v\text{ 与 }E'\text{ 的边关联}\}.$$

则子图 (V',E') 称为 G 的**由 E' 导出的子图**，简称为**边导出子图**，记为 $G[E']$. 若 $E'=\{e_{i_1},\cdots,e_{i_r}\}$，则

$$[e_{i_1},\cdots,e_{i_r}]\triangleq G[E'],$$

若从 G 中删去 E' 的所有的边，所得到的子图称为**删边子图**，记为 $G-E'$. 若 $E'=\{e\}$，则

$$G-e\triangleq G-\{e\}.$$

设 E' 是 G 中不相邻顶点的无序对所组成的边集，则 $G+E'$ 表示在 G 中添加 E' 的所有的边所得到的图，称为 G 的**加边图**. 若 $E'=\{e\}$，则

$$G+e\triangleq G+\{e\}.$$

容易证明(习题 1.8)：

$$G[V\backslash V']=G-V',\quad G[E\backslash E']\subseteq G-E'.$$

例 1.3 给定图 G 如图 1.8(*a*)，其点(边)导出子图，删点(边)子图与加边图如图 1.8(*b*) ~ (*f*).

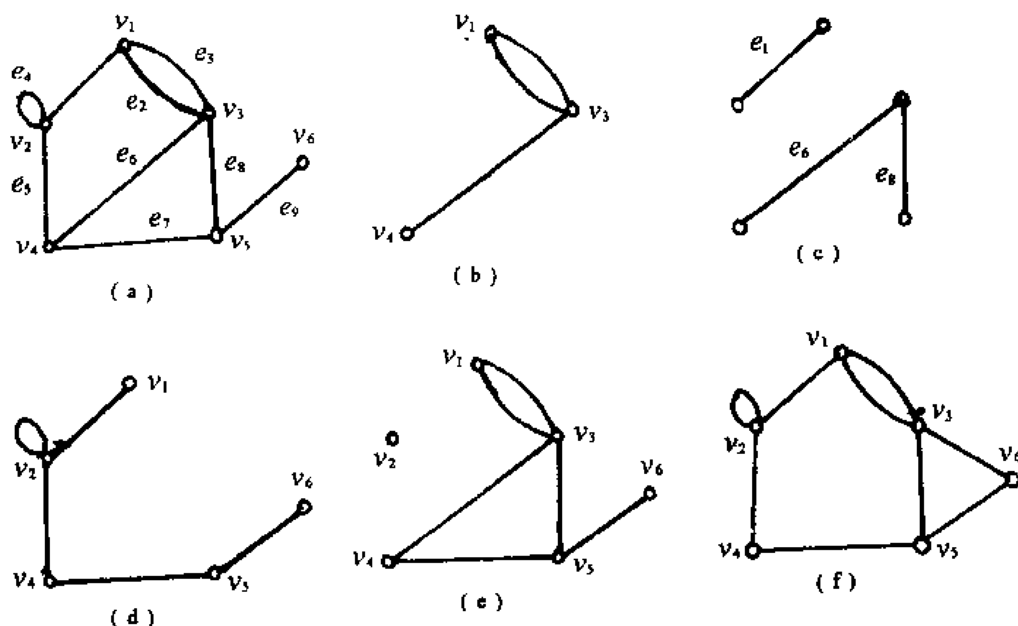


图 1.8 (a) 图 G　　(d) 删点子图 $G-v_3$

(b) 点导出子图$[v_1,v_3,v_4]$　　(e) 删边子图 $G-\{e_1,e_4,e_5\}$

(c) 边导出子图$[e_1,e_6,e_8]$　　(f) 加边图 $G+(v_3,v_6)$

设 G_1,G_2 是 G 的二个子图，若 $V(G_1)\cap V(G_2)=\varnothing$，则称 G_1 与 G_2 是**不相交的**. 若 $E(G_1)\cap E(G_2)=\varnothing$，则称 G_1 与 G_2 是**边不交的**. 不相交一定边不交，反之不一定.

定义 1.4(交)　设 G_1,G_2 是 G 的子图，则 G_1 与 G_2 的**交** $G_1\cap G_2$ 定义为

$$G_1\cap G_2\triangleq G[E(G_1)\cap E(G_2)].$$

若 G_1 与 G_2 边不交，则 $G_1\cap G_2$ 是空图.

定义 1.5(并)　设 G_1,G_2 是 G 的子图，则 G_1 与 G_2 的**并** $G_1\cup G_2$ 定义为

$$G_1\cup G_2\triangleq G[E(G_1)\cup E(G_2)].$$

若 G_1 与 G_2 边不交，则把 $G_1\cup G_2$ 记为 G_1+G_2.

定义 1.6(差)　图 G 与子图 H 的**差** $G-H$ 定义为

$$G-H\triangleq G[E(G)\backslash E(H)].$$

定义 1.7(补)　设 H 是 G 的子图，则 H 在 G 的**补图** $\overline{H}(G)$ 定义为

$$\overline{H}(G)\triangleq G-E(H).$$

若 G 是简单图，则 G 的补图 G^c 是与 G 有相同顶点集的一个简单图，且在 G^c 中两个顶点相邻当且仅当它们在 G 中不相邻.

定义 1.8(对称差)　设 G_1,G_2 是 G 的子图，则 G_1 与 G_2 的**对称差** $G_1\oplus G_2$ 定义为

$$G_1\oplus G_2\triangleq(G_1\cup G_2)-(G_1\cap G_2).$$

例 1.4　给定图 G 及子图 G_1,G_2 如图 1.9$(a)\sim(c)$，其交，并，差，对称差与补如图 1.9$(d)\sim(i)$.

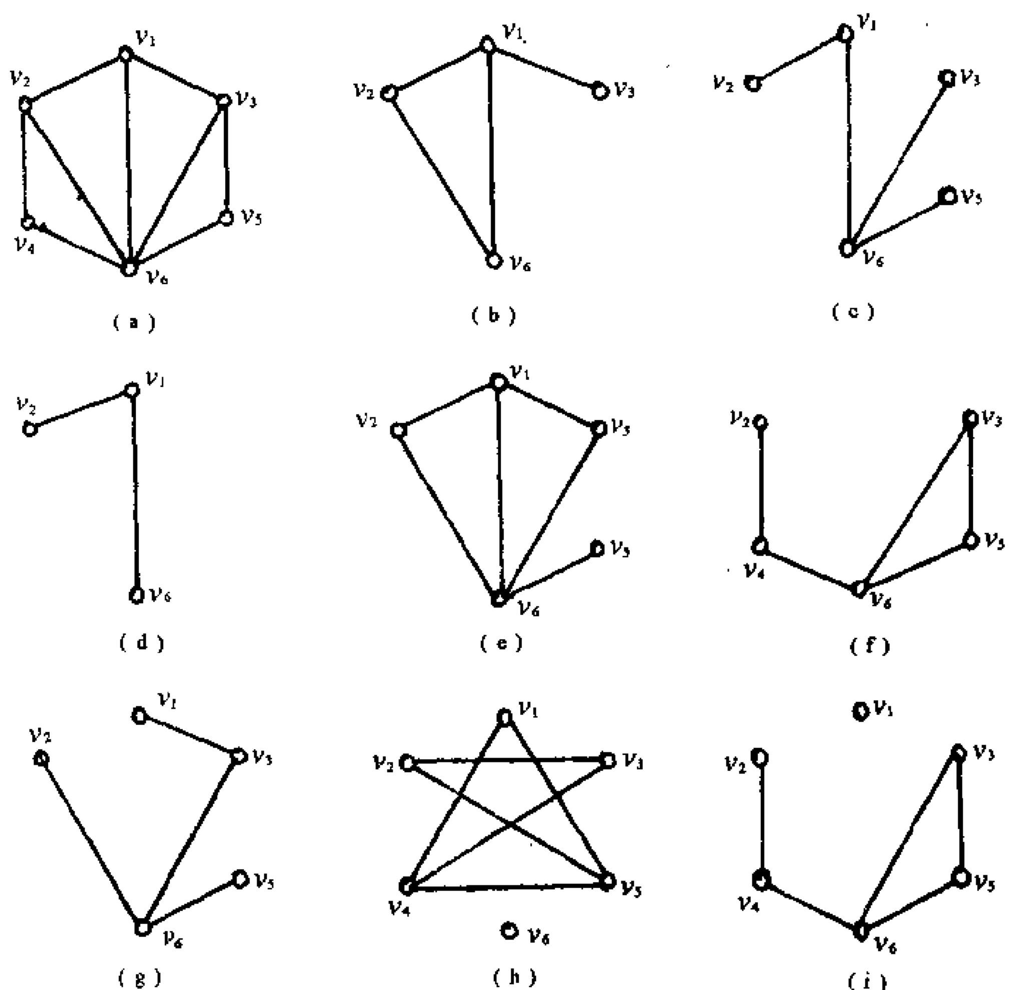


图1.9　(a) 图 G　(b) 子图 G_1　(c) 子图 G_2
(d) 交 $G_1 \cap G_2$　(e) 并 $G_1 \cup G_2$　(f) 差 $G - G_1$
(g) 对称差 $G_1 \oplus G_2$　(h) 补 G^c　(i) G_1 在 G 的补图 $\overline{G}_1(G)$

§1.4　有向图

定义 1.9　称 $G=(V,E)$ 为**有向图**，如果

(1) V 是一个非空有限集合，

(2) E 是 V 中元素的有序对所组成的有限集合，

并把 V 的元素叫做图的**顶点**，E 的元素叫做图的**有向边**或**边**.

设 G 是有向图，$e=(u,v)\in E(G)$，则称 u 为 e 的**起点**或**尾**，e 为 u 的**出边**；并称 v 为 e 的**终点**或**头**，e 为 v 的**入边**；又称 u 为 v 的**前趋**，v 为 u 的**后继**. 若二条或二条以上的边有相同的头和尾，则这些边称为**平行边**.

对有向图，顶点 v 的**前趋域** $\Gamma^-(v)$ 是

$$\Gamma^-(v) \triangleq \{u \in V(G) \mid u \text{ 是 } v \text{ 的前趋}\},$$

顶点 v 的**后继域** $\Gamma^+(v)$ 是

$$\Gamma^+(v) \triangleq \{u \in V(G) \mid u \text{ 是 } v \text{ 的后继}\},$$

顶点 v 的邻域 $\Gamma(v)$ 是

$$\Gamma(v) = \Gamma^+(v) \cup \Gamma^-(v).$$

例 1.5 对于简单有向图 $G=(V,E)$，可用映射 Γ^+ 或 Γ^- 表示，但不能用 Γ 表示. 对图 1.10 的有向图，可如下表示为 (V,Γ^+) 或 (V,Γ^-)：

$$\Gamma^+(v_1)=\{v_3,v_4,v_5\}, \qquad \Gamma^-(v_1)=\{v_2,v_3\},$$
$$\Gamma^+(v_2)=\{v_1,v_3\}, \qquad \Gamma^-(v_2)=\varnothing,$$
$$\Gamma^+(v_3)=\{v_1,v_4\}, \qquad \Gamma^-(v_3)=\{v_1,v_2\},$$
$$\Gamma^+(v_4)=\varnothing, \qquad \Gamma^-(v_4)=\{v_1,v_3,v_5\},$$
$$\Gamma^+(v_5)=\{v_4\}. \qquad \Gamma^-(v_5)=\{v_1\}.$$

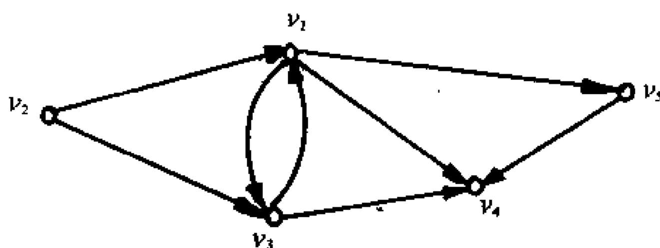


图 1.10 简单有向图

相对于有向图来说，定义 1.1 中的图称为**无向图**. 对于无向图，边 $e=(u,v)$ 也可以记为 $e=(v,u)$，因为无向图的边是顶点的无序对. 而有向图的边是顶点的有序对，故对有向图，(u,v) 与 (v,u) 是不同的二条边.

设 G 是有向图，若略去边的方向，则得到一个无向图，称为图 G 的**基础图**. 如果 G 是无向图，给所有的边任意定个方向，则得到一个有向图，称为图 G 的**定向图**. 容易看到，有向图的基础图是唯一的，而没有环的无向图的定向图却有 $2^{|E(G)|}$ 个.

例 1.6 给定有向图 G 如图 1.11(a)，其基础图如图 1.11(b). 对无向图(b)，它的一个定向图如(c).

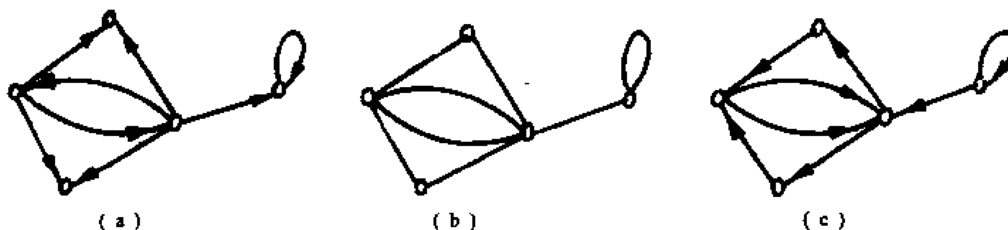


图 1.11 (a) 有向图 G (b) G 的基础图 G_1 (c) G_1 的定向图 G_2

设 G 是有向图，如果对任意的 $u,v \in V(G)$，以 u 为起点 v 为终点的边数，与 v 为起点 u 为终点的边数相同，则称 G 是**对称的**. 如果对任意的 $u,v \in G$，当 $(u,v) \in E(G)$ 时，则 $(v,u) \notin E(G)$，则称 G 为**反对称的**. 任意二顶点 u,v 都有二条边 (u,v) 和 (v,u) 的简单有向图称为**完全有向图**. 基础图是完全图的有向图称为**竞赛图**. 一个竞赛图也可看作是完全图的定向图.

例 1.7 在图 1.12 中给出对称图，反对称图，完全图和竞赛图的例.

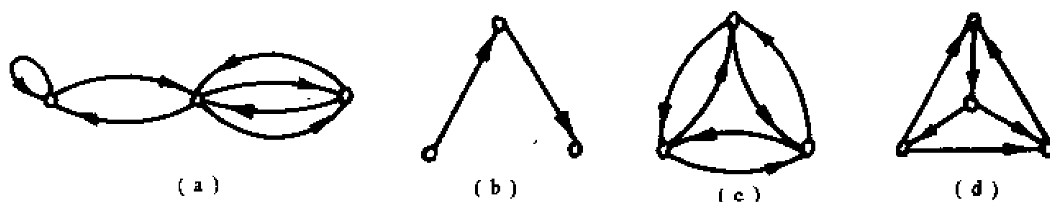


图 1.12 (a) 对称图 (b) 反对称图 (c) 完全图 (d) 竞赛图

§1.5 顶点度

定义 1.10 设 G 是一个图，每个 $v \in V(G)$，与 v 关联的边的数目称为 v 的**度**，记为 $d_G(v)$，简记为 $d(v)$，并规定在计算关联边数目时，环算作二条边.

我们用 $\delta(G)$ 或 δ，$\Delta(G)$ 或 Δ 分别表示图 G 的最小度，最大度.

孤立点不与任何边关联，故有

$$v \text{ 是孤立点} \iff d(v) = 0.$$

度为 1 的顶点称为**悬点**，与悬点关联的边称为**悬边**. 度为奇数的顶点称为**奇点**，度为偶数的顶点称为**偶点**. 每个顶点的度都相同的图称为**正则图**. 每个顶点的度都等于 k 的图称为 k **正则图**. 显然，完全图 K_n 是 $(n-1)$ 正则图，偶图 $K_{n,n}$ 是 n 正则图.

定理 1.1 对图 $G=(V,E)$，有

$$\sum_{v \in V} d(v) = 2|E|. \tag{1.1}$$

证 因为每条边有 2 个端点，换言之，在计算顶点的度时，每条边恰好重复计算一次，故式(1.1)成立. 证毕.

推论 1.2 图的奇点数目是偶数.

证 设图 $G=(V,E)$ 的奇点集为 V_1，偶点集为 V_2，则

$$\sum_{v \in V} d(v) = \sum_{v \in V_1} d(v) + \sum_{v \in V_2} d(v) = 2|E|,$$

因为 $\sum_{v \in V_2} d(v) =$ 偶数，$2|E| =$ 偶数，所以 $\sum_{v \in V_1} d(v)$ 也是偶数. 注意到 $v \in V_1$，$d(v)$ 是奇数，故 $|V_1|$ 是偶数. **证毕.**

对于有向图 G，顶点 v 的出边数称为**出度**，记为 $d^+(v)$. 顶点 v 的入边数称为**入度**，记为 $d^-(v)$. 顶点 v 的关联边数称为**度**，记为 $d(v)$，即

$$d(v) = d^+(v) + d^-(v), \quad v \in V(G).$$

我们用 $\delta^+(G)$ 或 δ^+，$\delta^-(G)$ 或 δ^-，$\delta(G)$ 或 δ 分别表示图 G 的最小出度，最小入度，最小度；用 $\Delta^+(G)$ 或 Δ^+，$\Delta^-(G)$ 或 Δ^-，$\Delta(G)$ 或 Δ 分别表示图 G 的最大出度，最大入度，最大度.

定理 1.3 对有向图 $G=(V,E)$，有

$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = |E|. \tag{1.2}$$

证　在有向图中，每条边恰好是一个顶点的出边，一个顶点的入边，故式(1.2)成立．证毕．

§1.6　连通性

定义 1.11　设 G 是一个图，G 的一个顶点和边的非空有限交错序列

$$W = v_0e_1v_1\cdots e_kv_k, \quad k \in N,$$

满足 $e_i = (v_{i-1}, v_i)$，$i = 1,2,\cdots,k$，则称 W 为一条 v_0-v_k **通道**，v_0 为**起点**，v_k 为**终点**，$v_1,\cdots,v_{k-1}$ 为**内顶点**，k 为通道的长．如果起点与终点不同，则称为**开通道**，若相同则称为**闭通道**．

若 W 是 v_0-v_k 通道，则从 W 的任意一个顶点 v_i 沿 W 到 W 上某个顶点 v_j 所得的通道称为 W 的 v_i-v_j **段**．把 W 的交错序列逆转得 v_k-v_0 通道，称为 W 的**逆通道**，记作 W^{-1}，即

$$W^{-1} = v_ke_k\cdots v_1e_1v_0.$$

没有重复边的通道称为**迹**．起点和终点不同的迹称为**开迹**．起点与终点重合的迹称为**闭迹**．

没有重复顶点的开通道称为**路**．

一条通道若顶点不重复则边必不重复，因此，一条路也是一条迹，但反过来不成立．

注意到一条通道中的顶点 v_{i-1} 与 v_i 是边 e_i 的两个端点，因此一条通道的点边交错序列可以略去顶点而用边序列表示．同样，迹和路也可用边序列表示．

如果 G 是简单图，则边由顶点对唯一确定．因此，在简单图中，通道、迹和路可用顶点序列表示．

例 1.8　给定图 1.13，有

开通道　$v_1e_1v_1e_2v_2e_2v_1e_3v_3$，

闭通道　$v_1e_1v_1e_2v_2e_2v_1$，

开　迹　$v_1e_1v_1e_3v_3e_4v_1e_2v_2$，

闭　迹　$v_1e_1v_1e_3v_3e_4v_1$，

路　　　$v_1e_2v_2e_6v_4e_7v_3$.

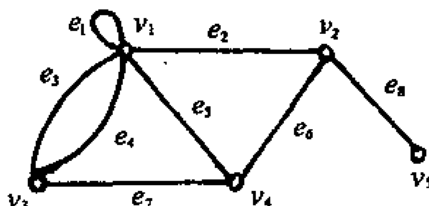


图 1.13

定理 1.4　图中若有 u-v 通道则必有 u-v 路．

证　设 W 是图中 u-v 通道．设想从起点 u 沿 W 旅行，一旦遇到已经到达过的顶点，例如 v_i，则从 W 中去掉 v_i-v_i 段，并从 v_i 开始沿 W 继续旅行，直到 W 的终点 v，由此即得一条没有重复顶点的 u-v 通道即 u-v 路．　证毕．

推论 1.5　在(n,m)图中若有 u-v 通道则必有其长不超过$(n-1)$的 u-v 路(习题 1.1)．

定义 1.12　G 的二个顶点 u,v 称为**连通的**，如果在 G 中有 u-v 通道．称图 G 是**连通的**，如果 G 的任二顶点都是连通的．

容易看到，图 G 的顶点间的连通关系是 $V(G)$ 的一个等价关系．于是，$V(G)$ 可按连通关系分解为等价类，设有 ω 个：$V_1,\cdots,V_\omega$，ω 是正整数．相应地，图 G 分解为 ω 个不相交子图，即

$$G = G[V_1] + \cdots + G[V_\omega],$$

称 $G[V_i]$ 为 G 的**连通支**或简称为**支**，$i = 1,\cdots,\omega$．并称 ω 为图 G 的**连通支数**．

由定义，图 G 的连通支就是图 G 的极大连通子图，因此，

$$G \text{ 连通 } \Longleftrightarrow \omega(G) = 1.$$

若 $u,v \in V(G)$ 连通，则有 u-v 通道. 一条最短 u-v 通道称为一条测地线，其长叫做顶点 u 与 v 间的**距离**，记作 $d(u,v)$. 显然测地线必为一条路. 若顶点 u,v 不连通，则规定其距离为 ∞. 容易证明由此所定义的顶点间的距离满足欧氏距离公理，即

(1) $d(u,v) \geqslant 0$，且 $d(u,v) = 0$ 当且仅当 $u = v$；（非负性）

(2) $d(u,v) = d(v,u)$；（对称性）

(3) $d(u,v) + d(v,w) \geqslant d(u,w)$. （三角不等式）

事实上，由定义，(1) 和 (2) 是明显的. 若 u 与 v 连通，v 与 w 连通，则 u 与 w 连通，且 u-v 通道加上 v-w 通道必为 u-w 通道，故 (3) 成立. 如果 u 与 v 不连通，或者 v 与 w 不连通，则

$$d(u,v) + d(v,w) = \infty,$$

故 (3) 仍成立. **证毕.**

图 G 的最长一条测地线的长称为图 G 的**直径**，记为 $d(G)$，即

$$d(G) = \max \{d(u,v) \mid u,v \in V(G)\}.$$

于是，G 连通当且仅当 $d(G) < \infty$.

在实际问题中通常不满足于知道图 G 是否连通，还需要知道图 G 的连通程度. 图的连通程度可作为系统可靠性的一种度量. 例如把图 1.9(b) 看作 4 个通讯站的一个通讯网络，若通讯站 v_1 发生故障中断工作时，整个通讯网络也将随之中断通讯. 这样的通讯网络，保证通讯畅通的可靠性就不高，或者说这样的图其连通程度不高. 为度量图的连通程度，引进连通度和边连通度的概念.

定义 1.13 设 G 连通，若 $V' \subseteq V(G)$ 使 $G - V'$ 不连通，则称 V' 是一个**顶点割**，有 k 个元素的顶点割称为 k **顶点割**. 不连通图的顶点割为空集.

图 G 的**连通度**是 G 的 k 顶点割中最小的 k，即使 G 不连通所需删去的最少顶点数，记为 $\kappa(G)$：

$$\kappa(G) = \begin{cases} \min \{|V'| \mid V' \text{ 是顶点割}\}, & G \text{ 至少有二个顶点不相邻}, \\ |V(G)| - 1, & \text{否则}. \end{cases}$$

若 $\kappa(G) \geqslant k$，则称 G 是 k **连通的**.

显然，非平凡图 G 不连通，当且仅当 $\kappa(G) = 0$.

设 S,T 是 $V(G)$ 的子集，以 $[S,T]$ 表示 S 与 T 之间的边集，即一个端点在 S 中，另一端点在 T 中的所有边的集.

定义 1.14 设 $\varnothing \neq S \subset V(G)$，$\overline{S} = V(G) \setminus S$，则称 $[S,\overline{S}]$ 为图 G 的一个**边割**. 边割 $[S,\overline{S}]$ 也记为 $N(S)$. 有 k 个元素的边割称为 k **边割**.

图 G 的**边连通度**是 G 的 k 边割中最小的 k，即使 G 不连通所需删去的最少边数，记为 $\lambda(G)$：

$$\lambda(G) = \begin{cases} 0, & G \text{ 平凡图或不连通}, \\ \min \{E' \mid E' \text{ 是边割}\}, & \text{否则}. \end{cases}$$

若 $\lambda(G) \geqslant k$，则称 G 是 k **边连通的**.

例 1.9 在图 1.14 中给出具有不同连通度和边连通度的图.

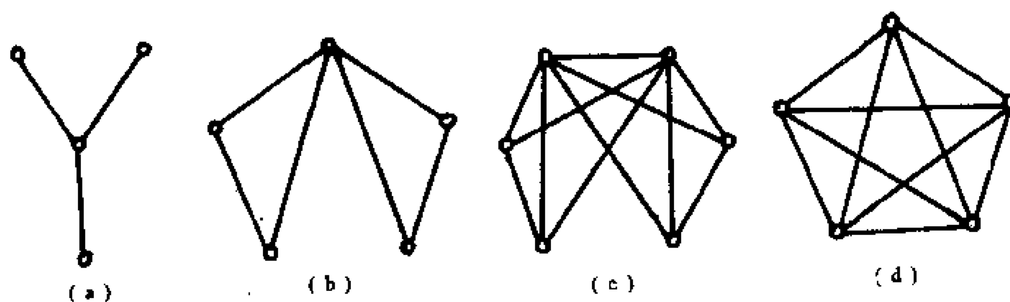


图 1.14　(a) 1 连通，1 边连通　　(b) 1 连通，2 边连通
(c) 2 连通，3 边连通　　(d) 4 连通，4 边连通

定理 1.6　对任何图 G 都有

$$\kappa(G) \leqslant \lambda(G) \leqslant \delta(G). \tag{1.3}$$

证　若 G 是平凡图或不连通图，则

$$\kappa(G) = \lambda(G) = 0,$$

故式(1.3)成立. 今考虑非平凡连通图 G，则与度为 δ 的顶点相关联的边构成 G 的一个 δ 边割，故 $\lambda(G) \leqslant \delta$.

设 G' 是 G 的基图，则

$$\kappa(G') = \kappa(G), \quad \lambda(G') \leqslant \lambda(G). \tag{1.4}$$

若能证明 $\kappa(G') \leqslant \lambda(G')$，则从式(1.4)推出式(1.3)左边不等式成立.

设 G 是完全图，则

$$\kappa(G) = \lambda(G) = |V(G)| - 1.$$

以上讨论说明，为证明 $\kappa(G) \leqslant \lambda(G)$，不妨设 G 是非平凡简单连通图且不是完全图. 假设 $F = [S, \bar{S}]$ 是一个边割，合于

$$|F| = \lambda(G) \geqslant 1,$$

则 F 的真子集不再是边割，故 $G - F$ 恰有二个连通支，设为 G_1, G_2. 如图 1.15，设与 F 的边关联的顶点在 G_1 中是 $x_1, \cdots, x_k$，在 G_2 中是 $y_1, \cdots, y_l$. 令

$$X = \{x_1, \cdots, x_k\}, Y = \{y_1, \cdots, y_l\}.$$

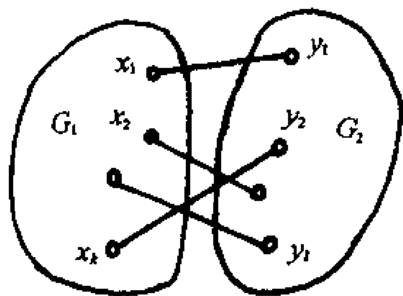


图 1.15

若 $V(G_1)\backslash X \neq \varnothing$，则 X 是 G 的顶点割且

$$\kappa(G) \leqslant |X| = k \leqslant |F| = \lambda(G).$$

于是我们可以假定

$$G_1 = G[X], \quad G_2 = G[Y],$$

$$\because \quad d_G(x_i) \geqslant \delta \geqslant |F|, \quad \sum_{i=1}^{k} d_F(x_i) = |F|,$$

$$\therefore \quad d_G(x_i) - d_F(x_i) \geqslant \sum_{j \neq i} d_F(x_j) \geqslant k - 1,$$

因此 $G[X]$ 是完全图. 于是每个 $x_i \in X$ 在 G_2 中至少与 $(|F| - k + 1)$ 个顶点相邻，故从 G_1 的顶点到 G_2 的顶点的边数至少有 $k(|F| - k + 1)$ 条，所以 $k(|F| - k + 1) \leqslant |F|$，即

$$(k - |F|)(k - 1) \geqslant 0,$$

故 $k = 1$ 或 $|F|$. 同理,$G[Y]$ 是完全图,$l = 1$ 或 $|F|$. 因为 G 不是完全图,所以

$$k = l = |F| \geqslant 2,$$

即 $G[X]$ 与 $G[Y]$ 都是 $|F|$ 个顶点的完全图且它们之间恰有 $|F|$ 条互不相邻的边,故 G 是 $|F|$ 正则图. 任取 $v \in V(G)$,与 v 相邻的 $|F|$ 个顶点构成一个顶点割,故

$$\kappa(G) \leqslant |F| = \lambda(G).$$

从而定理得证.

读者可以对任意的正整数 $n \leqslant m \leqslant l$,作出一个简单图 G 使 $\kappa(G) = n, \lambda(G) = m, \delta(G) = l$(习题 1.33).

例 1.10　取 $\kappa = 2, m = 4, \delta = 4$,可以作出如图 1.16 的简单图,合于 $\kappa = 2, \lambda = 4, \delta = 4$.

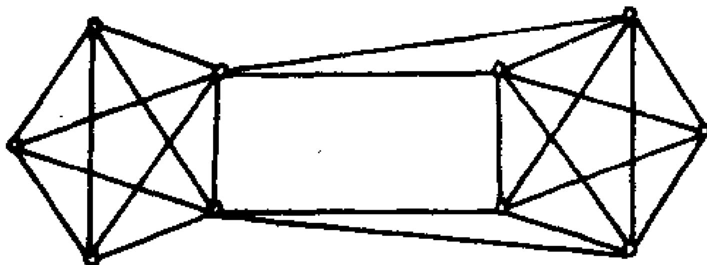


图 1.16

关于有向图的连通性,比无向图的情形要复杂.

定义 1.15　设 $G = (V,E)$ 是有向图,G 的一个顶点和边的非空有限交错序列

$$W = v_0e_1v_1\cdots e_kv_k$$

满足 $e_i = (v_{i-1}, v_i), i = 1,\cdots,k$,则称 W 为一条 v_0-v_k **通道**. 若 $v_0 \neq v_k$,则称为**有向开通道**. 若 $v_0 = v_k$,则称为**有向闭通道**. 边不重复的有向通道称为**有向迹**. 顶点不重复的有向开通道称为**有向路**.

如果有一条 u-v 有向通道,则称 u 到 v **可达**. 若 u 到 v 可达或 v 到 u 可达,则称顶点 u 与 v 是**单向连通的**. 若 u 到 v 可达且 v 到 u 可达,则称顶点 u 与 v 是**强连通的**.

称有向图 G 是**连通的**,如果 G 的基础图是连通的. 称 G 是**单向连通的**,如果 G 的任二顶点都是单向连通的. 称 G 是**强连通的**,如果 G 的任二顶点都是强连通的.

显然,强连通必单向连通,单向连通必连通,但反过来不成立.

例 1.11　如图 1.17 的三个有向图,说明连通不一定单向连通,单向连通不一定强连通.

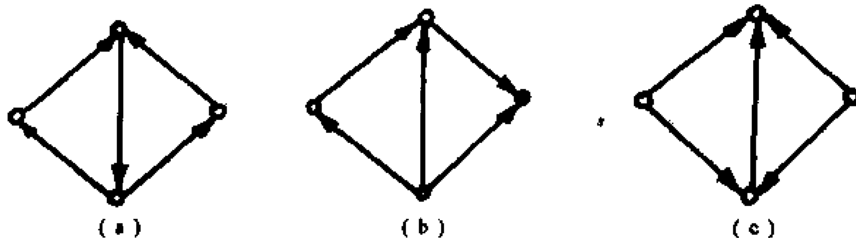


图 1.17 (a) 强连通,单向连通,连通 (b) 非强连通,单向连通,连通 (c) 非强连通,非单向连通,连通

对有向图 G，顶点间的连通关系，强连通关系都是等价关系，但单向连通关系不是等价关系. 按连通关系对有向图顶点集的分类实际上就是对有向图的基础图进行的. 假定对有向图 G 的顶点集 V 按强连通关系分解为 $V_1,\cdots,V_{\omega}$，则 $G[V_i]$ 称为有向图 G 的**强连通支**，$i=1,\cdots,\omega$. 有向图 G 的强连通支就是 G 的极大强连通子图. 类似地我们可以定义有向图 G 的**单向连通支**为 G 的极大单向连通子图.

例 1.12 给定有向图 G 如图 1.18(a)，其强连通支 5 个如(b_1) ~ (b_5)，单向连通支 3 个如(c_1) ~ (c_3)，连通支 2 个如(d_1) ~ (d_2). G 的强连通支互不相交，且包含了 G 的所有顶点而不一定包含 G 的所有的边. G 的单向连通支可能有公共顶点，也可能有公共边. G 的连通支互不相交，且包含了 G 的所有顶点和边.

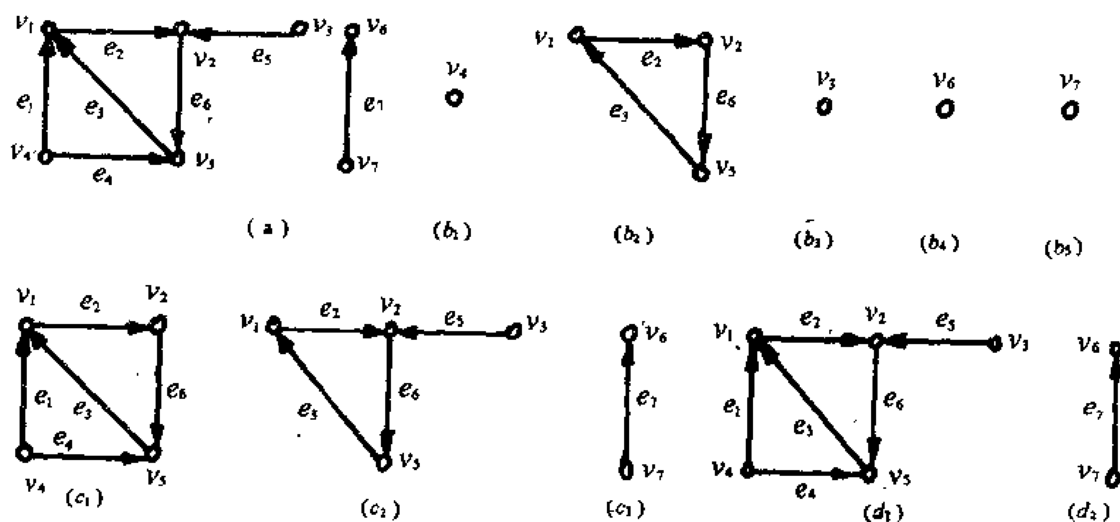


图 1.18

设 $G=(V,E)$ 是有向图，容易证明(习题 1.25)：

(1) 每个顶点恰好在一个强连通支，每条边或者不在任何强连通支或者恰在一个强连通支；

(2) 每个顶点和每条边都在某个单向连通支；

(3) 每个顶点和每条边恰在一个连通支.

在有向图中顶点 u 到顶点 v 的距离定义为最短 u-v 有向通道的长. 如果不存在 u-v 有向通道，则规定 $d(u,v)=\infty$. 由此所定义的距离满足非负性和三角不等式，但不满足对称性.

§1.7 圈和余圈

定义 1.16 起点和终点重合的路称为圈. 长为 k 的圈称为 k 圈. 长为奇数的圈称为奇圈. 长为偶数的圈称为偶圈.

由定义，圈就是内顶点不重复的闭通道.

定理 1.7 闭迹可分解为边不交的圈之并.

证 从闭迹上某顶点出发沿闭迹旅行，当到达已经经过的顶点时，得到一个圈. 从该顶点出发沿闭迹上没有旅行过的边继续旅行，当旅行终止时，已分解闭迹为边不交的圈之

并．　证毕．

由定理 1.7 可知，圈就是极小闭迹．

定理 1.8　若 $\delta \geqslant 2$，则图必有圈．

证　不妨设 G 是连通图，否则可考查图 G 的任一连通支．任取 $v_0 \in V(G)$，从 v_0 出发沿 G 的边旅行，其规则是：沿没有经过的边到达没有经过的顶点，否则旅行终止．由于 $\delta \geqslant 2$，所以在旅行中，当到达某个顶点 v_i 时，必能离开 v_i 继续旅行．由于顶点数有限，旅行必终止在已经经过的某个顶点，于是得到一个圈．　**证毕．**

定理 1.9　若 $|E(G)| \geqslant |V(G)|$，则 G 含圈．

证　对顶点数归纳证明．若 $|V(G)| = 1$，则 $|E(G)| \geqslant 1$，即 G 有环，定理自然成立．假设当 $|V(G)| = n$ 时定理成立，今设 $|V(G)| = n + 1$．如果每个顶点度 $\geqslant 2$，由定理 1.8，G 必有圈．否则取 $v_0 \in V(G)$，$d(v_0) = 0$ 或 1．于是

$$|E(G - v_0)| \geqslant |V(G - v_0)| = n,$$

由归纳假设，$G - v_0$ 含圈，该圈也是 G 的圈，故得证．

定理 1.10　图 G 是偶图，当且仅当 G 不含奇圈．

证　**必要性**　设 $G = (X, Y, E)$ 是偶圈，并设

$$C = v_0 v_1 \cdots v_k v_0$$

是圈，$v_0 \in X$．因为 G 是偶圈，所以依次有

$$v_1, v_3, v_5, \cdots, v_k \in Y,$$

故 k 是奇数．

充分性　不妨设 G 连通，取 $v_0 \in V(G)$，设

$$X = \{x \in V(G) \mid d(x, v_0) \text{ 偶数}\},$$

$$Y = \{y \in V(G) \mid d(y, v_0) \text{ 奇数}\}.$$

则 X, Y 是 $V(G)$ 的一个分解．设 $v, u \in X$，P 是最短 v_0-v 路，Q 是最短 v_0-u 路．今从 v_0 出发沿 P 旅行，与 Q 的最后一个公共顶点是 w(也许 $w = v_0$)，并设 P_1 为 P 的 w-v 段，Q_1 为 Q 的 w-u 段，如图 1.19．因为 P 是最短路，所以 P 的 v_0-w 段是最短 v_0-w 路．同理，Q 的 v_0-w 段也是最短 v_0-w 路，故它们的长相等．又因为 P 与 Q 的长都是偶数，所以 v-u 路 $P_1^{-1} + Q_1$ 的长也是偶数．因为图 G 不含奇圈，所以顶点 v 与顶点 u 不相邻．于是我们已证得 X 的任二顶点都不相邻．同理，Y 的任二顶点也不相邻，故 G 是偶圈．　**证毕．**

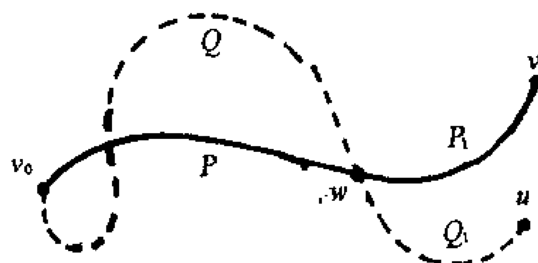


图 1.19

定义 1.17　图 G 的极小边割称为余圈．

定理 1.11　设 G 连通，$F \subseteq E(G)$，则

$$F \text{ 是余圈 } \Longleftrightarrow \omega(G - F) = 2.$$

证　F 是余圈 $\Longleftrightarrow F$ 是边割，F 的真子集都不是边割

$\Longleftrightarrow G - F$ 不连通，且不存在 F 的真子集 F' 使 $G - F'$ 不连通

$\Longleftrightarrow \omega(G - F) = 2$．　**证毕．**

推论 1.12 设 G 连通，$\varnothing \neq S \subset V(G)$，则

$$[S,\overline{S}] \text{ 是余圈 } \Longleftrightarrow G[S], G[\overline{S}] \text{ 都连通}.$$

证 注意到 $G - [S,\overline{S}] = G[S] + G[\overline{S}]$，故

$$\begin{aligned}[S,\overline{S}] \text{ 是余圈} &\Longleftrightarrow \omega(G - [S,\overline{S}]) = 2 \\ &\Longleftrightarrow \omega(G[S] + \omega[\overline{S}]) = 2 \\ &\Longleftrightarrow \omega(G[S]) = \omega(G[\overline{S}]) = 1 \\ &\Longleftrightarrow G[S], G[\overline{S}] \text{ 都连通}.\end{aligned}$$

证毕.

定理 1.13 边割可分解为边不交余圈之并.

证 设 $[S,\overline{S}]$ 是边割，$G - S$ 的连通支是 $G_1,\cdots,G_k$，如图 1.20(a)，则

$$[S,\overline{S}] = [V(G_1),\overline{V(G_1)}] + \cdots + [V(G_k),\overline{V(G_k)}].$$

为证明定理，只需证明：每个 $[V(G_i),\overline{V(G_i)}]$ 可分解为边不交余圈之并.

设 C 是包含 G_i 的连通支，$C - V(G_i)$ 的连通支是 $C_1,\cdots,C_l$，如图 1.20(b)，则

$$[V(G_i),\overline{V(G_i)}] = [V(C_1),\overline{V(C_1)}] + \cdots + [V(C_l),\overline{V(C_l)}].$$

对每个 j，$G[V(C_j)] = C_j$ 连通，$G[\overline{V(C_j)}] = C - C_j$ 连通，由推论 1.12，$[V(C_j),\overline{V(C_j)}]$ 是余圈. 证毕.

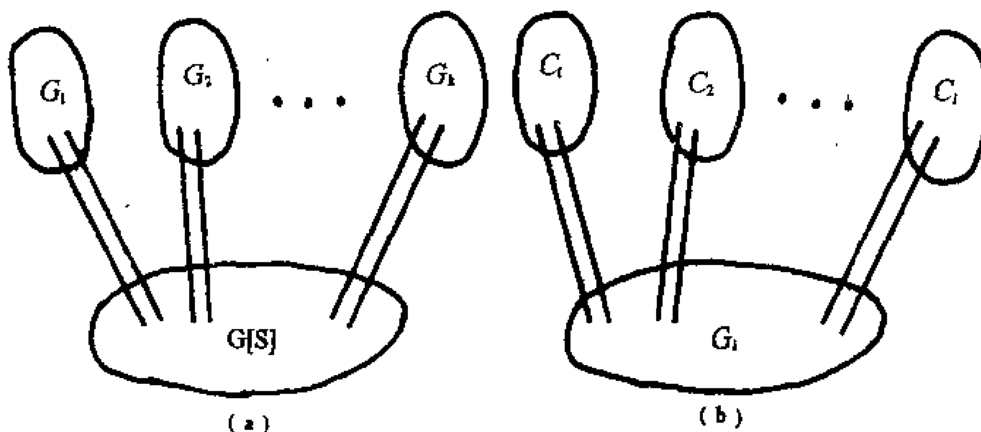


图 1.20 (a) $G - S$ 的分解　　(b) $C - V(G_i)$ 的分解

对于有向图的圈和余圈，定义为它的基础图的圈和余圈. 有向图的圈若圈上边的方向都一致则称为**有向圈**. 有向图的余圈若余圈上边的方向都一致则称为**有向余圈**.

定理 1.14 设 $G = (V,E)$ 是有向图，$E = \{1,2,\cdots,m\}$，若边 1 着黑色，其余的边或者不着色，或者任意着红、黑、绿三色之一，则恰有下列情形之一出现：

(1) 存在含边 1 的圈，圈上不含未着色的边且黑边与边 1 同向，绿边与边 1 反向；

(2) 存在含边 1 的余圈，余圈上不含红边且黑边与边 1 同向，绿边与边 1 反向.

证 相继给顶点以如下标记：

(a) 设边 $1 = (t,s)$，标记 s；

(b) 若 x 已标记，y 未标记，且

1) 存在黑边 (x,y)，或者

2) 存在红边 (x,y) 或 (y,x)，或者

3) 存在绿边 (y,x)；

则标记 y.

当标记终止时恰有下列情形之一出现.

情形 1 t 被标记.

此时,从标记 s 到标记 t 有一条 s-t 迹,加上边 1 得一闭迹. 由标记规则,此闭迹由红、黑、绿边组成且黑边方向与边 1 相同,绿边方向与边 1 相反. 分解此闭迹为边不交的圈之并,含边 1 的圈即为定理之(1) 所求. 再证在此情形不会兼有定理之(2) 出现. 不然,若有余圈 $[S,\overline{S}]$ 合于定理之(2),不妨设 $s\in S,t\in\overline{S}$. 根据从 $s\in S$ 到 $t\in\overline{S}$ 的标记过程,必能从某个 $x\in S$ 标记到 $y\in\overline{S}$,如图 1.21. 有三种可能:

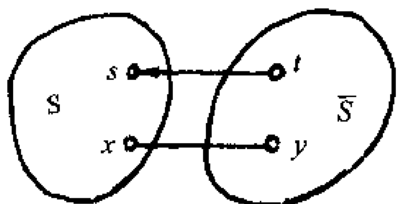


图 1.21　从 x 标记到 y

1°) (x,y) 是黑边;

2°) (x,y) 或 (y,x) 是红边;

3°) (y,x) 是绿边;

无论哪一种可能都与定理之(2) 矛盾.

情形 2 t 未被标记.

设 $S=\{v|v$ 被标记$\}$ 则 $s\in S,t\in\overline{S}$. 因为 $\overline{S}$ 的顶点在标记过程中都是不能作标记的,所以边割 $[S,\overline{S}]$ 不含与边 1 反向的黑边,不含红边,也不含与边 1 同向的绿边. 分解此边割为边不交的余圈之并,含边 1 的余圈即为定理之(2) 所求. 再证在此情形不会兼有定理之(1) 出现. 不然,若有圈 C 合于定理之(1). 因为 s 已作标记,t 未作标记,所以,在圈 C 上从 t 出发,沿边 1 相反的方向旅行,当第一次遇到已标记顶点 u 时终止,并设最后离开的一个未标记顶点是 v,如图 1.22,故 $u\in S,v\in\overline{S}$. 根据标记过程,不能从 u 标记到 v,有三种可能:



图 1.22　不能从 u 标记到 v

1°) (v,u) 是黑边;

2°) (u,v) 或 (v,u) 是未着色的边;

3°) (u,v) 是绿边;

无论哪一种可能都与定理之(1) 矛盾.

定理**证毕**.

推论 1.15　有向图的每条边或恰在一个有向圈上,或恰在一个有向余圈上.

证　由定理 1.14,所有的边都着黑色,故得证.

定理 1.16　设 G 连通有向图,$E(G)\neq\varnothing$,则下列各条等价:

(1) G 强连通;

(2) 每条边都在一个有向圈上;

(3) G 不含有向余圈.

证　(1) ⇒ (2)　设 $(x,y)\in E(G)$,由强连通性,存在 y-x 有向路 P,故 $P+(x,y)$ 是含 (x,y) 的有向圈.

(2) ⇒ (3)　由推论 1.15.

(3) ⇒ (1)　若 G 不是强连通的，则 G 的强连通支至少有 2 个，设为 G_1, G_2. 因为 G 连通，故有边 (x,y) 连接 G_1 与 G_2，不妨设 $x \in V(G_1), y \in V(G_2)$. 因为 x 与 y 在不同的强连通支，所以边 (x,y) 不在任何有向圈上. 由推论 1.15，(x,y) 在某个有向余圈上，此与 G 不含有向余圈矛盾.

定理证毕.

§1.8　图的矩阵表示

一个图可以用图形来表示，并且常常把图与表示图的图形等同起来. 用图形表示图的优点是直观、方便. 若用矩阵表示图则便于计算机处理.

定义 1.18　(n,m) 图 G 的关联矩阵 $A(G)$ 是 $n \times m$ 矩阵，其 i 行 j 列的元 a_{ij} 定义为

$$a_{ij} = \begin{cases} 0, & e_j \text{ 与 } v_i \text{ 不关联}, \\ 1, & e_j \text{ 与 } v_i \text{ 关联且 } e_j \text{ 不是环}, \\ 2, & e_j \text{ 是 } v_i \text{ 的环}. \end{cases} \quad i = 1,2,\cdots,n;\ j = 1,2,\cdots,m.$$

例 1.13　如图 1.13 的图 G，其关联矩阵是

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{1.5}$$

关联矩阵 A 有如下性质：

(1) 考查 A 的列，有

e_j 是环 $\Longleftrightarrow A_{\cdot j}$ 有一个元是 2 其余都是 0，

e_j 不是环 $\Longleftrightarrow A_{\cdot j}$ 有二个元是 1 其余都是 0，

e_j 与 e_k 是平行边 $\Longleftrightarrow A_{\cdot j} = A_{\cdot k}$，

$\sum\limits_{i=1}^{n} a_{ij} = 2,\ j = 1,2,\cdots,m$；

(2) 考查 A 的行，有

v_i 是孤立点 $\Longleftrightarrow a_{ij} = 0,\ j = 1,2,\cdots,m$，

$\sum\limits_{j=1}^{m} a_{ij} = d(v_i),\ i = 1,2,\cdots,n$，

v_i 是悬点 $\Longleftrightarrow A_{i\cdot}$ 有一个元是 1 其余都是 0；

(3) $A(G)$ 依赖于顶点和边的顺序，并且

顶点顺序的改变 $\Longleftrightarrow A(G)$ 相应的行变换，

边顺序的改变 $\Longleftrightarrow A(G)$ 相应的列变换.

换言之，设 A 是 G 的关联矩阵，则 A' 是 G 的关联矩阵的充要条件是存在置换矩阵 P，Q 使

$$A' = PAQ.$$

例 1.14 图 1.13 的图 G 的关联矩阵可写为

$$A' = \begin{array}{c} \\ v_2 \\ v_4 \\ v_3 \\ v_5 \\ v_1 \end{array}\begin{array}{c} \begin{array}{cccccccc} e_2 & e_4 & e_7 & e_8 & e_1 & e_3 & e_5 & e_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

因为有置换矩阵

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

使 $A' = PAQ$，这里 A 即式(1.5) 的关联矩阵. 实际上 A' 的顶点顺序是 $v_2v_4v_3v_5v_1$，边的顺序是 $e_2e_4e_7e_8e_1e_3e_5e_6$.

(4) 图 G 有 ω 个连通支 $G_1,\cdots,G_\omega$ 的充要条件是，适当安排顶点和边的顺序，使 $A(G)$ 具有分块对角阵的形式，即

$$A(G) = \begin{pmatrix} A(G_1) & & \\ & \ddots & \\ & & A(G_\omega) \end{pmatrix}.$$

(5) 矩阵 A 是一个无环图的关联矩阵，当且仅当

(a) A 的元素为 0 或 1；(b) A 的每列恰有 2 个"1".

定理 1.17 设 A 是 (n,m) 图 G 的关联矩阵，则在模 2 运算下，$r(A) = n - \omega$.

证 只需证明当 G 连通时 $r(A) = n - 1$.

(1) A 至多 $(n-1)$ 个独立行，因为 A 的列恰有 2 个 1，其所有行的和必为零.

(2) A 的任意 $(n-1)$ 行独立. 不然，设有

$$A_{i_1\cdot} + A_{i_2\cdot} + \cdots + A_{i_r\cdot} = 0, \quad (1 \leqslant r \leqslant n-1). \tag{1.6}$$

因为 G 连通，必有

$$e = (v_{i_t}, v) \in E(G), v \neq v_{i_1}, \cdots, v_{i_r},$$

于是 $A_{i_1\cdot} + A_{i_2\cdot} + \cdots + A_{i_r\cdot}$ 相应于边 e 的分量是 1，此与式(1.6) 矛盾.

由(1) 与(2)，定理得证.

设 G 是连通 (n,m) 图，则 $r(A) = n - 1$，且 A 的任意 $(n-1)$ 行是独立的. 我们称 A 的任意 $(n-1)$ 行构成的子矩阵为**基本关联矩阵**，记为 A_f. 特别地，当去掉 A 的顶点 v_i 的行所得到的基本关联矩阵记为 $A_{f,i}$.

对无环图的关联矩阵 A，任意 $(n-1)$ 行都是独立的. 设 A_1 是从 A 任意略去一行的子矩

阵,利用关联矩阵每列恰有两个 1 的性质,可以把略去的行还原.

定义 1.19 (n,m) 图 G 的邻接矩阵 $H(G)$ 是 $n\times n$ 矩阵,其 i 行 j 列的元 h_{ij} 定义为

$$h_{ij}=\text{以 } v_i \text{ 与 } v_j \text{ 为端点的边数}\qquad i,j=1,2,\cdots,n.$$

例 1.15 如图 1.13 的图 G,其邻接矩阵是

$$H=\begin{array}{c}\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\end{array}\begin{array}{c}\begin{array}{ccccc}v_1 & v_2 & v_3 & v_4 & v_5\end{array}\\ \begin{pmatrix}1 & 1 & 2 & 1 & 0\\ 1 & 0 & 0 & 1 & 1\\ 2 & 0 & 0 & 1 & 0\\ 1 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\end{pmatrix}\end{array}. \tag{1.7}$$

邻接矩阵 H 有如下性质:

(1) 邻接矩阵是对称阵;

(2) 在顶点 v_i 有环 $\Longleftrightarrow h_{ii}>0$,

在顶点 v_i 与 v_j 有平行边 $\Longleftrightarrow h_{ij}>1$,

G 是简单图 $\Longleftrightarrow H(G)$ 是对角线为零的 0-1 矩阵;

(3) $$d(v_i)=2h_{ii}+\sum_{\substack{k=1\\k\neq i}}^{n}h_{ik}=2h_{ii}+\sum_{\substack{k=1\\k\neq i}}^{n}h_{ki},$$

若 G 是简单图,则 $d(v_i)=H_{i\cdot}$ 中"1"的个数 $=H_{\cdot i}$ 中"1"的个数;

(4) $H(G)$ 依赖于顶点的顺序,并且

顶点顺序的改变 $\Longleftrightarrow A(G)$ 相应行和列的交换.

换言之,设 H 是 G 的邻接矩阵,则 H' 是 G 的邻接矩阵的充要条件是存在置换矩阵 P 使

$$H'=P^{-1}HP.$$

例 1.16 图 1.13 的图 G 的邻接矩阵可写为

$$H'=\begin{pmatrix}0 & 1 & 0 & 1 & 1\\ 1 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 0 & 2\\ 1 & 0 & 0 & 0 & 0\\ 1 & 1 & 2 & 0 & 1\end{pmatrix}.$$

因为有置换矩阵

$$P=\begin{pmatrix}0 & 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\end{pmatrix},$$

使 $H'=P^{-1}HP$,这里 H 是式(1.7)的邻接矩阵.实际上 H' 的顶点顺序是 $v_2v_4v_3v_5v_1$.

(5) G 是偶图的充要条件是,适当安排顶点的顺序使 $H(G)$ 有如下形式:

$$H(G)=\begin{pmatrix}0 & K\\ K^T & 0\end{pmatrix}.$$

(6) 图 G 有 ω 个连通支 $G_1,\cdots,G_\omega$ 的充要条件是，适当安排顶点的顺序使 $H(G)$ 具有分块对角阵的形式，即

$$H(G)=\begin{pmatrix} H(G_1) & & \\ & \ddots & \\ & & H(G_\omega) \end{pmatrix}.$$

定理 1.18 设 H 为 (n,m) 图 G 的邻接矩阵，并记 $H^k=(h_{ij}^{(k)})$，k 为正整数，则 $h_{ij}^{(k)}$ 等于 G 的 v_i-v_j 通道数.

证 当 $k=1$ 时定理自然成立. 设定理对 H^k 成立，今证对 H^{k+1} 也成立. 事实上由

$$H^{k+1}=H\cdot H^k$$

得

$$h_{ij}^{(k+1)}=\sum_{t=1}^{n} h_{it}h_{tj}^{(k)},$$

其中

$$h_{it}=\text{长为 1 的 } v_i\text{-}v_t \text{ 通道数},$$
$$h_{tj}^{(k)}=\text{长为 } k \text{ 的 } v_t\text{-}v_j \text{ 通道数},$$

故

$$h_{it}h_{tj}^{(k)}=\text{长为}(k+1)\text{的过 } v_t \text{ 的 } v_i\text{-}v_j \text{ 通道数}.$$

于是 $h_{ij}^{(k+1)}=$ 长为 $(k+1)$ 的 v_i-v_j 通道数. **证毕.**

推论 1.19 设 H 是 (n,m) 图 G 的邻接矩阵，令

$$L^{(k)}\triangleq H+H^2+\cdots+H^k\triangleq(l_{ij}^{(k)}),\quad k=1,2,\cdots,n-1. \tag{1.8}$$

则

(1) 当 $i\neq j$，$l_{ij}^{(k)}=$ 长小于等于 k 的 v_i-v_j 通道数；

(2) 当 $i=j$，$l_{ij}^{(k)}=$ 长小于等于 k 的 v_i-v_j 闭通道数；

(3) 当 $i\neq j$，v_i 与 v_j 连通的充要条件是 $l_{ij}^{(n-1)}>0$.

证 (1) 与(2) 是定理 1.18 的直接结果. 今证(3).

v_i 与 v_j 连通 $\Longleftrightarrow$ 存在 v_i-v_j 通道

$\Longleftrightarrow$ 存在长小于等于 $(n-1)$ 的 v_i-v_j 通道

$\Longleftrightarrow l_{ij}^{(n-1)}>0$.

故得证.

推论 1.20 设 H 是 (n,m) 图 G 的邻接矩阵，则

$$d(v_i,v_j)=\min\{k\mid h_{ij}^{(k)}\neq 0\},\ i\neq j. \tag{1.9}$$

证 直接由定理得证.

例 1.17 已知图 1.13 的图 G 的邻接矩阵由式(1.7) 给出，则有

$$H=\begin{pmatrix} 1 & 1^* & 2^* & 1^* & 0 \\ 1^* & 0 & 0 & 1^* & 1^* \\ 2^* & 0 & 0 & 1^* & 0 \\ 1^* & 1^* & 1^* & 0 & 0 \\ 0 & 1^* & 0 & 0 & 0 \end{pmatrix},\qquad H^2=\begin{pmatrix} 7 & 2 & 3 & 4 & 1^* \\ 2 & 3 & 3^* & 1 & 0 \\ 3 & 3^* & 5 & 2 & 0 \\ 4 & 1 & 2 & 3 & 1^* \\ 1^* & 0 & 0 & 1^* & 1 \end{pmatrix},$$

$$H^3 = \begin{pmatrix} 19 & 12 & 18 & 12 & 2 \\ 12 & 3 & 5 & 8 & 3 \\ 18 & 5 & 8 & 11 & 3^* \\ 12 & 8 & 11 & 7 & 1 \\ 2 & 3 & 3^* & 1 & 0 \end{pmatrix}, \qquad H^4 = \begin{pmatrix} 79 & 33 & 50 & 49 & 12 \\ 33 & 23 & 32 & 20 & 3 \\ 50 & 32 & 47 & 31 & 5 \\ 49 & 20 & 31 & 31 & 8 \\ 12 & 3 & 5 & 8 & 3 \end{pmatrix},$$

$$L^{(4)} = H + H^2 + H^3 + H^4 = \begin{pmatrix} 106 & 48 & 73 & 66 & 15 \\ 48 & 29 & 40 & 30 & 7 \\ 73 & 40 & 60 & 45 & 8 \\ 66 & 30 & 45 & 41 & 10 \\ 15 & 7 & 8 & 10 & 4 \end{pmatrix}.$$

在 H^t 中打"＊"号的元素表示满足式(1.9),于是

$$d(v_1,v_2) = d(v_1,v_3) = d(v_1,v_4) = d(v_2,v_4) = d(v_2,v_5) = d(v_3,v_4) = 1,$$

$$d(v_1,v_5) = d(v_2,v_3) = d(v_4,v_5) = 2,$$

$$d(v_3,v_5) = 3.$$

由 $l_{11}^{(4)} = 106$,得长小于等于 4 的 v_1-v_1 闭通道有 106 条,由 $l_{35}^{(4)} = 8$ 得长小于等于 4 的 v_3-v_5 通道有 8 条.

下面讨论有向图的矩阵表示.

定义 1.20 无环 (n,m) 有向图 G 的**关联矩阵** $A(G)$ 是 $n \times m$ 矩阵,其 i 行 j 列的元 a_{ij} 定义为

$$a_{ij} = \begin{cases} 0, & e_j \text{ 与 } v_i \text{ 不关联}, \\ 1, & e_j \text{ 是 } v_i \text{ 的出边}, \\ -1, & e_j \text{ 是 } v_i \text{ 的入边}. \end{cases} \quad i = 1,\cdots,n;\ j = 1,\cdots,m.$$

例 1.18 如图 1.23 的有向图 G,其关联矩阵是

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{pmatrix} 1 & 1 & -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array}.$$

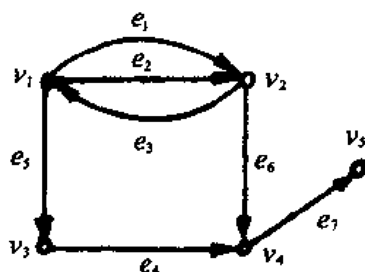


图 1.23

因为每条边恰好是一个顶点的出边,另一个顶点的入边,所以有向图的关联矩阵每列恰好一个"1",一个"-1",其余元素都是零.

关联矩阵的 i 行,其"1"的个数为顶点的出度,"-1"的个数为顶点的入度,非零元素个数为顶点的度.

类似于定理 1.17 可证,(n,m) 连通有向图的关联矩阵的秩为 $(n-1)$,且任意 $(n-1)$ 行是独立的.

定义 1.21 (n,m) 有向图 G 的**邻接矩阵** $H(G)$ 是 $n\times n$ 矩阵，其 i 行 j 列的元 h_{ij} 定义为

$$h_{ij}=\text{从 } v_i \text{ 到 } v_j \text{ 的边数}, \qquad i,j=1,2,\cdots,n.$$

例 1.19 如图 1.24 的有向图 G，其邻接矩阵是

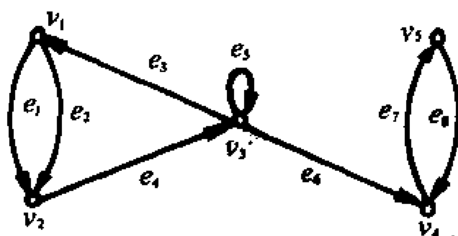


图 1.24

$$H=\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ v_1 & 0 & 2 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 0 & 0 \\ v_3 & 1 & 0 & 1 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 0 & 0 & 1 & 0 \end{matrix}.$$

由定义即知，H 的行其各元之和为顶点的出度，H 的列其各元之和为顶点的入度.

对有向图 G 的邻接矩阵 H，类似于定理 1.18 可得：

$h_{ij}^{(k)}=$ 长为 k 的 v_i-v_j 有向通道数，

$l_{ij}^{(k)}=$ 长小于等于 k 的 v_i-v_j 有向通道数.

定义 1.22 (n,m) 有向图 G 的**可达矩阵** $P(G)$ 是 $n\times n$ 矩阵，其 i 行 j 列的元 p_{ij} 定义为

$$p_{ij}=\begin{cases}1, & v_i \text{ 到 } v_j \text{ 可达},\\ 0, & \text{否则}.\end{cases} \qquad i,j=1,2,\cdots,n.$$

距离矩阵 $D(G)$ 是 $n\times n$ 矩阵，其 i 行 j 列的元 d_{ij} 定义为

$$d_{ij}=d(v_i,v_j), \qquad i,j=1,2,\cdots,n.$$

由于顶点 v_i 到自身是可达的，故 $p_{ii}=1$. 当 $i\neq j$ 时有

$$v_i \text{ 可达 } v_j \Longleftrightarrow \text{存在长小于等于}(n-1)\text{的 } v_i\text{-}v_j \text{ 有向通道}$$
$$\Longleftrightarrow l_{ij}^{(n-1)}\neq 0.$$

于是

$$p_{ij}=\begin{cases}1, & l_{ij}^{(n-1)}\neq 0 \text{ 或 } i=j,\\ 0, & \text{否则},\end{cases} \tag{1.10}$$

$$d_{ij}=\begin{cases}0, & i=j,\\ \infty, & l_{ij}^{(n-1)}=0, i\neq j,\\ \min\{k \mid h_{ij}^{(k)}\neq 0\}, & \text{否则}.\end{cases} \tag{1.11}$$

例 1.20 对图 1.24 的有向图 G，有

$$H=\begin{pmatrix}0 & 2^* & 0 & 0 & 0\\ 0 & 0 & 1^* & 0 & 0\\ 1^* & 0 & 1 & 1^* & 0\\ 0 & 0 & 0 & 0 & 1^*\\ 0 & 0 & 0 & 1^* & 0\end{pmatrix}, \qquad H^2=\begin{pmatrix}0 & 0 & 2^* & 0 & 0\\ 1^* & 0 & 1 & 1^* & 0\\ 1 & 2^* & 1 & 1 & 1^*\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix},$$

$$H^3=\begin{pmatrix}2&0&2&2^*&0\\1&2&1&1&1^*\\1&2&3&2&1\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix},\qquad H^4=\begin{pmatrix}2&4&2&2&2^*\\1&2&3&2&1\\3&2&5&4&2\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix},$$

$$L^{(4)}=H+H^2+H^3+H^4=\begin{pmatrix}4&6&6&4&2\\3&4&6&4&2\\6&6&10&8&4\\0&0&0&2&2\\0&0&0&2&2\end{pmatrix}.$$

在 H^k 中打"*"的元素表示满足式(1.9)，故由式(1.10)与(1.11)得

$$P=\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix},\qquad D=\begin{pmatrix}0&1&2&3&4\\2&0&1&2&3\\1&2&0&1&2\\\infty&\infty&\infty&0&1\\\infty&\infty&\infty&1&0\end{pmatrix}.$$

例 1.21 在无向图中，若顶点 v_i 与 v_j 连通，则可看作 v_i 可达 v_j 且 v_j 可达 v_i. 故式(1.10)与式(1.11)对无向图也是适用的，只是应注意到在无向图中可达矩阵和距离矩阵是对称的.

对图 1.13 的无向图 G，由例 1.17，其可达矩阵和距离矩阵是

$$P=\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{pmatrix},\qquad D=\begin{pmatrix}0&1&1&1&2\\1&0&2&1&1\\1&2&0&1&3\\1&1&1&0&2\\2&1&3&2&0\end{pmatrix}.$$

§1.9 图的同构

定义 1.23 称图 G 与图 H **同构**，记为 $G\cong H$，如果

(1) 存在 $V(G)$ 与 $V(H)$ 的一一对应；

(2) 存在 $E(G)$ 与 $E(H)$ 的一一对应；

(3) 在(1)与(2)的一一对应保持顶点与边的关联关系不变.

图到自身的同构称为**自同构**.

当略去一个图的顶点与边的标号所得到的图称为**非标定图**. 对一个非标定图的顶点与边任意给以标号所得到的图称为**标定图**. 两个图同构即它们的非标定图相同. 反过来，给定一个非标定图，其任意两个标定图都是同构的. 换言之，凡同构的图，可以用一个非标定图来代表.

例 1.22 4 个顶点的所有不同构的简单图，其非标定图共有 11 个，如下表所示.

$\lvert E(G)\rvert$	非标定图
0	
1	
2	
3	
4	
5	
6	

定理 1.21 两个简单图 G_1, G_2 同构的充要条件是

(1) 存在 $V(G_1)$ 与 $V(G_2)$ 的一一对应,并且

(2) 保持顶点间的相邻关系不变.

证 因为简单图的边由顶点的相邻关系唯一决定,故得证.

定理 1.22 设 H_1 与 H_2 分别是图 G_1 与 G_2 的邻接矩阵,则 $G_1 \cong G_2$ 的充要条件是存在置换矩阵 P 使

$$H_1 = P^{-1}H_2P.$$

证 G_1 与 G_2 同构即 G_1 与 G_2 有相同的非标定图.对一个非标定图的不同标号即对邻接矩阵相应行与列的置换,故得证.

设 G 是简单图,由定理 1.21 和定理 1.22,G 的自同构就是 $V(G)$ 上保持相邻关系的置换.并且,保持相邻关系的两个置换的乘积仍是保持相邻关系的置换.保持相邻关系的置换的逆仍是保持相邻关系的置换.因此,G 的置换所构成的集合,在置换的乘法下是一个群,称为 G 的**自同构群**,记作 $A(G)$.

例 1.23 $A(K_n)$ 是 n 阶对称群,即

$$A(K_n) = \left\{ \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix} \middle| i_1, i_2, \cdots, i_n \text{ 是 } 1, 2, \cdots, n \text{ 的一个排列} \right\}.$$

例 1.24 3 个顶点 $V = \{1,2,3\}$ 的简单图共有 4 个,如图 1.25 所示.其自同构群有 2 个:

$$A(G_1) = A(G_4) = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ i_1 & i_2 & i_3 \end{pmatrix} \middle| i_1, i_2, i_3 \text{ 是 } 1,2,3 \text{ 的一个排列} \right\},$$

$$A(G_2) = A(G_3) = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \right\}.$$

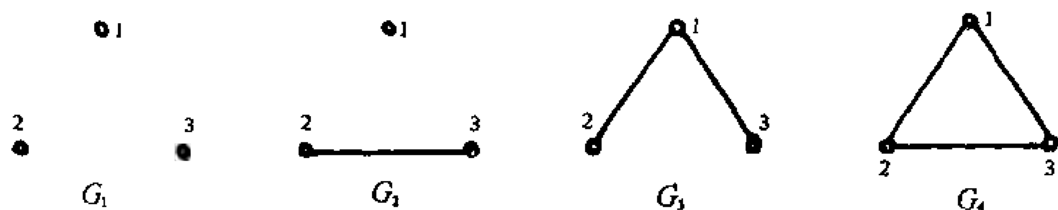


图 1.25　3 个顶点的简单图

如何判断两个图是否同构至今还没有一个好算法. 两个图 G_1 与 G_2 同构的必要条件可罗列一些如下：

(1) $|V(G_1)| = |V(G_2)|$，　　(2) $|E(G_1)| = |E(G_2)|$，

(3) $\omega(G_1) = \omega(G_2)$，　　(4) $\sigma(G_1) = \sigma(G_2)$，

(5) $\kappa(G_1) = \kappa(G_2)$，　　(6) $\lambda(G_1) = \lambda(G_2)$，

(7) 如果把图的顶点度由大到小排成一个序列，称为度序列，则 G_1 与 G_2 有相同的度序列，

(8) $H(G_1)$ 与 $H(G_2)$ 有相同的特征多项式.

关于有向图的同构类似于无向图，只是应注意到边是有向的.

例 1.25　如图 1.26 的 4 个有向图，(a) 与 (b) 同构，因为有同构对应

$$a \longleftrightarrow 2,\quad b \longleftrightarrow 1,\quad c \longleftrightarrow 4,\quad d \longleftrightarrow 3,$$

而(c) 与(d) 不同构，因为在悬点一个只有出边，一个只有入边.

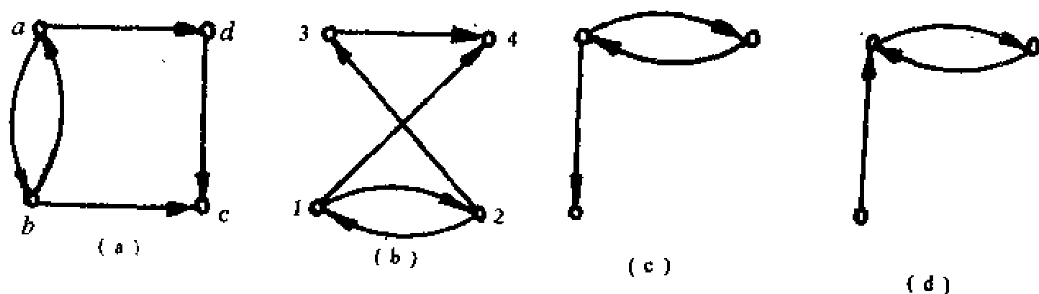


图 1.26　有向图的同构

习　题　一

1.1　设 G 是一个图，$v \in V(G)$，证明

$$\text{在顶点 } v \text{ 有环} \Longleftrightarrow v \in \Gamma(v).$$

1.2　图 G 是零图的充要条件是，对每个 $v \in V(G)$，$\Gamma(v) = \varnothing$.

1.3　简单图 G 是完全图，当且仅当

$$\Gamma(v) = V(G)\backslash\{v\} \quad \forall\, v \in V(G).$$

1.4　简单 k- 分图是完全 k- 分图，当且仅当 $\forall\, i$，$\Gamma(V_i) = V(G)\backslash V_i$.

1.5　设 G 是(n,m) 图

(1) 若 G 是简单图，则 $m \leqslant \binom{n}{2}$，

(2) G 是完全图的充要条件是 $m = \binom{n}{2}$.

1.6 证明

(1) G 是简单偶图，则 $|E(G)| \leqslant \frac{|V(G)|^2}{4}$，

(2) $G \triangleq K_{r,s}$ 是完全偶图，则 $|E(K_{r,s})| = rs$.

1.7 设顶点集

$$V = \{(\delta_1, \cdots, \delta_k) | \delta_i = 0 \text{ 或 } 1, i = 1, \cdots, k\},$$

V 的两个顶点相邻当且仅当它们恰有一个坐标不同，由此构成的图称为 k- **立方**，记为 Q_k. 证明 Q_k 是有 2^k 个顶点与 $k \cdot 2^{k-1}$ 条边的偶图，并画出 Q_1, Q_2 与 Q_3.

1.8 设图 $G = (V, E), \varnothing \neq V' \subseteq V, \varnothing \neq E' \subseteq E$，证明

(1) $G[V \backslash V'] = G - V'$，

(2) $G[E \backslash E'] \subseteq G - E'$，

并举例说明 $G[E \backslash E'] \neq G - E'$.

1.9 完全图的点导出子图仍是完全图.

1.10 偶图的子图仍是偶图.

1.11 设 G_1, G_2 是 G 的两个子图，证明

$$G_1 \oplus G_2 = G[E_1 \oplus E_2],$$

其中 $E_1 = E(G_1), E_2 = E(G_2)$.

1.12 假定认识是相互的，证明

(1) 任意 6 个人，总有 3 人互相认识，或者总有 3 人互不认识，

(2) 任意 9 个人，总有 3 人互相认识，或者总有 4 人互不认识，

(3) 任意 18 个人，总有 4 人互相认识，或者总有 4 人互不认识.

1.13 任意 n 个人的人群($n \geqslant 2$)，至少有 2 人，他们在该人群中朋友数相同，假定朋友是相互的.

1.14 设 $S = \{x_1, \cdots, x_n\}$ 是平面点集，且任二点的距离大于等于 1，证明距离为 1 的点对最多 $3n$ 对.

1.15 证明 $\delta(G) \leqslant \frac{2|E(G)|}{|V(G)|} \leqslant \Delta(G)$.

1.16 在 (n,m) 图中若有 u-v 通道，则有长度不超过 $(n-1)$ 的 u-v 路.

1.17 对简单图必有长大于等于 δ 的路.

1.18 $\forall\ e \in E(G)$，$\omega(G) \leqslant \omega(G - e) \leqslant \omega(G) + 1$.

1.19 对简单 (n,m) 图，证明

$$n - \omega \leqslant m \leqslant \frac{1}{2}(n - \omega)(n - \omega + 1).$$

1.20 设有 n 个电话交换台，问每个电话交换台至少与另外多少个电话交换台有直通线路就能使其互通电话，为什么？

1.21 若 n 个电话交换台可互通电话，则至少要多少条直通线路，为什么？

1.22　设 G 是 n 个顶点的简单图，若 G 不含三角形，则

$$|E(G)| \leqslant \left[\frac{n^2}{4}\right],$$

并作出不含三角形的 $\left(n,\left[\frac{n^2}{4}\right]\right)$ 图.

1.23　设 G 是 (n,m) 简单图，$n > 1$

(1) 若 $m > \binom{n-1}{2}$，则 G 连通，

(2) 举例 $m = \binom{n-1}{2}$ 且 G 不连通.

1.24　设 G 是简单图，则 G 或者 G^c 恰有一个是连通的.

1.25　设 $G = (V,E)$ 是有向图，证明

(1) 每个顶点恰在一个强连通支，每条边或者不在任何强连通支或者恰在一个强连通支，

(2) 每个顶点和每条边都在某个单向连通支，

(3) 每个顶点和每条边都恰在一个连通支.

1.26　若 $d(G) > 3$，则 $d(G^c) < 3$.

1.27　平面上给定 $2n$ 个点，今有一些圆，每个圆至少包含 $n+1$ 个点，试证这 $2n$ 个点的任二点连线必整个含于某些圆内.

1.28　设 G' 是连通图 G 的子图，$V(G') \subset V(G)$，则有

$$e \in E(G) \backslash E(G'),$$

e 的一个端点在 G' 而另一个端点不在 G'.

1.29　连通图的任意二条最长的路必相交.

1.30　设 G 无环，若 $\forall\ v \in V(G)$，$d(v) \geqslant 3$，则 G 必含偶圈.

1.31　设 G 连通，则 G 是圈的充要条件是

$$d(v) = 2,\ v \in V(G).$$

1.32　设 G 连通，则 G 是路的充要条件是 G 有 2 个悬点面其余顶点度都等于 2.

1.33　对任意的正整数

$$l \leqslant m \leqslant n,$$

作出简单图 G 使 $\kappa = l, \lambda = m, \delta = n$.

1.34　设 G 是 k- 边连通的，则

$$|E(G)| \geqslant \frac{k|V(G)|}{2}.$$

1.35　设 G 是简单图，$0 < k < |V(G)| - 1$，若

$$\delta \geqslant \frac{|V(G)| + k - 2}{2},$$

则 G 是 k- 连通的.

1.36　简单有向图是强连通的，当且仅当 G 有一条包含每个顶点的有向闭通道.

1.37　简单有向图必有长大于等于

$$\max\ \{\delta^+, \delta^-\}$$

的有向路.

1.38　设 G 是简单有向图，

$$\max\{\delta^+,\delta^-\}=k>0,$$

则 G 包含长大于等于 $(k+1)$ 的有向圈.

1.39　若有向图 G 没有有向圈，则

$$\delta^+=\delta^-=0.$$

1.40　设 G 是有向图，则 G 没有有向圈的充要条件是对 G 的每个顶点 v 可给以标号 $l(v)$，使

$$l(u)<l(v),\ (u,v)\in E(G).$$

1.41　设 G 是竞赛图，则存在顶点 v，从 v 可沿长为 1 或 2 的有向路到达其余每个顶点.

1.42　设 G 是竞赛图，$|V(G)|\geqslant 3$，则 G 有生成有向圈或只改变一条边的方向就得到生成有向圈.

1.43　设 G 是强连通的 (n,m) 竞赛图，$n\geqslant 3$，则每个顶点都在一个有向 k 圈上 $(3\leqslant k\leqslant n)$.

1.44　竞赛图是强连通的，当且仅当有生成有向圈.

1.45　设 G 是竞赛图，若 $\varnothing\neq S\subset V(G)$ 且 S 的出边集 $N^+(S)=\varnothing$，则称 S 为**孤立集**. 证明 G 有生成有向圈的充要条件是 G 没有孤立集.

1.46　设 G 是有向图，强连通支是 $G_1,\cdots,G_\omega$. 作有向图 G^*，顶点 $u_1,\cdots,u_\omega$ 对应 ω 个强连通支，且 $(u_i,u_j)\in E(G^*)$ 当且仅当 G 中有边从 $V(G_i)$ 到 $V(G_j)$，则称 G^* 为 G 的**凝聚图**. 证明有向图的凝聚图没有有向圈.

1.47　设 G^* 是竞赛图 G 的凝聚图，$|V(G^*)|=k$，则 G^* 的顶点可如下排列：

$$v_1^*,v_2^*,\cdots,v_k^*,$$

使得对每个 $i,1\leqslant i\leqslant k,G^*$ 中头为 v_i^* 的所有的边，其尾都在 $\{v_1^*,v_2^*,\cdots,v_{i-1}^*\}$ 中.

1.48　设 H 是 (n,m) 图 G 的邻接矩阵，I 是单位矩阵，则 G 连通的充要条件是

$$(H+I)^{n-1}>0.$$

1.49　设 (n,m) 图 G 无环，A 是关联矩阵，则

$$(AA^T)_{ij}=\begin{cases}\text{以 } v_i,v_j \text{ 为端点的边数}, & i\neq j,\\ d(v_i), & i=j.\end{cases}$$

1.50　矩阵 A 是一个无环连通图的关联矩阵. 当且仅当 A 是 0-1 矩阵，且 A 的每列恰有两个 1.

1.51　设 G 是简单图，证明 $\Lambda(G)=\Lambda(G^c)$.

1.52　构造一个非平凡简单图，其自同构群仅含单位变换.

1.53　设 G 是简单图，若 $G\cong G^c$，则称 G 为自补图. 证明对 (n,m) 自补图满足

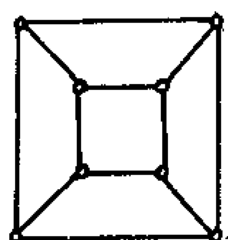
$$n\equiv 0\quad\text{or }1\ (\text{mod }4).$$

1.54　找出 4 个顶点的自补图.

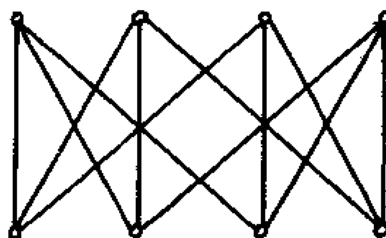
1.55　设 G 是简单图，邻接矩阵的特征值各异，证明 $\Lambda(G)$ 是 Abel 群.

1.56　设 $G_1\cong G_2$，则 G_1 与 G_2 的邻接矩阵的特征多项式相同.

1.57　证明图 1.27(a) 与(b) 的两个图是同构的.



(a)

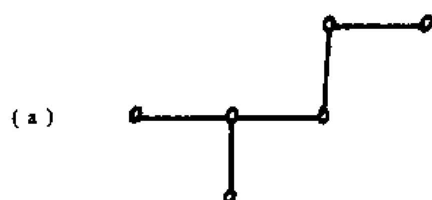


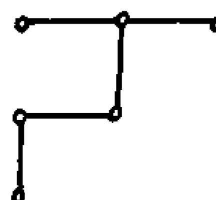
(b)

图 1.27

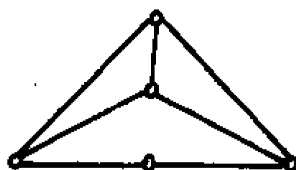
1.58　图 1.28 的 4 对图是否同构,为什么?

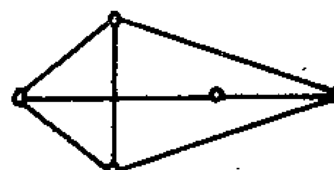
(a)



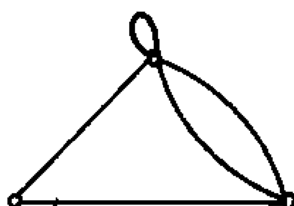


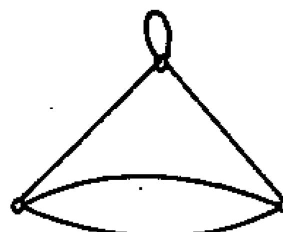
(b)



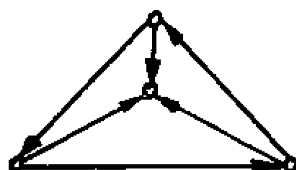


(c)





(d)



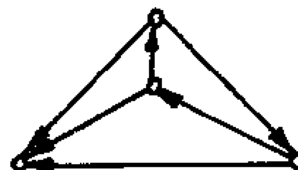


图 1.28

第二章　树及其应用

§2.1　树

定义 2.1　无圈连通图称为**树**.树中最长路的长称为树的**高**.树的度为 1 的顶点称为**树叶**,其余的顶点称为**分枝点**.树的边称为**树枝**.

无圈图称为**森林**.

由定义,森林即每个连通支都是树的图.树的树叶就是悬点.

例 2.1　具有 4 个顶点的树的非标定图只有两个,如图 2.1.

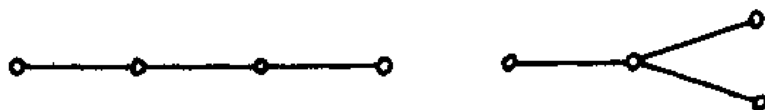


图 2.1　4 个顶点的树

定理 2.1　设 T 是非平凡 (n,m) 图,下列命题等价:

(1) T 是树;　　(2) T 无圈且 $m=n-1$;

(3) T 连通且 $m=n-1$;　　(4) T 无圈,任加一边有唯一圈;

(5) T 连通,任删一边不再连通;　　(6) T 的任二顶点恰有一条路连通.

证　(1) ⇒ (2)　当 $n=2$,$T=K_2$,结论自然成立.设 $n\geqslant k(\geqslant 2)$ 结论成立.今设 $n=k+1$.因 T 连通且无圈,由定理 1.8,T 必有悬点,设为 v_0,则 $T-v_0$ 是 $(n-1,m-1)$ 图且是树.由归纳假设,$m-1=(n-1)-1$,即 $m=n-1$.

(2) ⇒ (3)　设 T 有 ω 个连通支:$T_1,\cdots,T_\omega$,因为 T 无圈,故每个 T_i 都是树.设 T_i 是 (n_i,m_i) 图,则

$$m_i=n_i-1,\ i=1,\cdots,\omega,$$

于是

$$m=\sum_{i=1}^{\omega}m_i=\sum_{i=1}^{\omega}(n_i-1)=\sum_{i=1}^{\omega}n_i-\omega=n-\omega,$$

由条件 $m=n-1$ 得 $\omega=1$,故 T 连通.

(3) ⇒ (4)　当 $n=2$,$T=K_2$,结论自然成立.设 $n\geqslant k(\geqslant 2)$ 结论成立.今设 $n=k+1$.若 $\delta\geqslant 2$,则

$$2m=\sum_{i=1}^{n}d(v_i)\geqslant 2n,$$

此与 $m=n-1$ 矛盾.再由连通性,T 必有悬点,设为 v_0,则 $T-v_0$ 是 $(n-1,m-1)$ 图且满足(3).由归纳假设,$T-v_0$ 无圈而任加一边有唯一圈.因为 v_0 是悬点所以 v_0 不在任何圈

上，故 T 无圈. 在 T 任加一边 (u,v)，有 2 种情形：

1) $u,v \in V(T-v_0)$. 此时 (u,v) 为加在 $T-v_0$ 的边，故 $T+(u,v)$ 有唯一圈，

2) $u \in V(T-v_0)$，$v=v_0$. 不妨设悬边是 (u_0,v_0). 因为 u,u_0 在 $T-v_0$ 连通，故加边 (u,v_0) 得 $T+(u,v)$ 的唯一圈.

(4) ⇒ (5)　若 T 不连通则存在 $u,v \in V(T)$ 不连通，于是加边 (u,v) 后没有圈，矛盾，故 T 连通. 若存在 $e \in E(G)$ 使 $T-e$ 连通则 T 含圈，矛盾，故 T 任删一边不再连通.

(5) ⇒ (6)　若顶点 u,v 间有两条路连通则必含圈，删去圈上的边仍连通，矛盾.

(6) ⇒ (1)　由条件(6)，T 连通且无圈，故 T 是树. 至此定理得证.

推论 2.2　非平凡树至少有二片树叶.

证　设 T 非平凡树，k 是树高. 不妨设路 P 的长为 k. 因为 P 是最长路，故 P 的起点 u 与终点 v 都不能与 P 外的顶点相邻. 因为 T 无圈，故也不能有边 $e \in E(T)\backslash E(P)$ 使 u 或 v 与 P 的顶点相邻. 所以 $d(u)=d(v)=1$.　**证毕.**

若图 G 的生成子图 T 是树，则称 T 为**生成树**.

定理 2.3　图 G 连通当且仅当 G 有生成树.

证　树是连通的，故充分性得证. 今证必要性. 设 G 连通，则有两种情形：

(1) 任删 G 的一条边不再连通，

(2) 删去 G 的某条边仍连通.

在情形(1)，由定理 2.1，G 是树. 在情形(2)，得 G 的连通生成子图 G_1.

以 G_1 代 G 继续上述过程，由于边数有限，故在有限步之后必然出现情形(1)，即得 G 的生成树.　**证毕.**

§2.2　分离点和桥

定义 2.2　设 $v_0 \in V(G)$，若 $E(G)$ 有一个分解 E_1,E_2 使

$$V(G[E_1]) \cap V(G[E_2]) = v_0, \tag{2.1}$$

则称 v_0 是 G 的一个**分离点**.

例 2.2　对无环图，至少有 3 个顶点的连通图才有分离点，如图 2.2(a) ~ (c). 对有环图，1 个顶点的图也可能有分离点，如图 2.2(d) ~ (f).

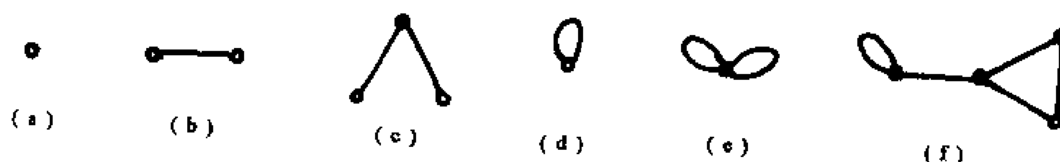


图 2.2　图的分离点(用实心点表示)

定理 2.4　设 G 无环，则 $v_0 \in V(G)$ 是分离点的充要条件是

$$\omega(G-v_0) \geqslant \omega(G)+1. \tag{2.2}$$

证　必要性　v_0 是分离点，不妨设 $(v_0,v_1) \in E_1$，$(v_0,v_2) \in E_2$. 若存在 v_1-v_2 路不含 v_0，则 v_0 在圈上，此与式(2.1)矛盾. 因此 v_1-v_2 路必含 v_0，故 v_1 与 v_2 在 $G-v_0$ 的不同连通支中，即式(2.2)成立.

充分性　设 v_0 在 G 的连通支 H 中，并设

$$H - v_0 = H_1 + H_2 + \cdots + H_k (k \geqslant 2),$$

令

$$E_1 = E(H_1) \cup \{(v_0, v) \in E(H) | v \in V(H_1)\},$$

则 E_1 与 $E_2 \triangleq E(G) \backslash E_1$ 为满足式(2.1)的一个分解．　证毕．

从定理 2.4，对无环图，分离点就是只有一个元素的顶点割．

定理 2.5　设 G 连通无环图，$v_0 \in V(G)$，下列命题等价：

(1) v_0 是分离点；

(2) $G - v_0$ 不连通；

(3) 存在 $V(G) \backslash \{v_0\}$ 的一个分解 V_1, V_2，对任意 $w \in V_1$ 和 $u \in V_2$，v_0 在每一条 w-u 路上；

(4) 存在异于 v_0 的顶点 w 与 u，v_0 在每一条 w-u 路上．

证　(1) 与(2) 的等价性由定理 2.4.

(1) ⇒ (3)　因为 G 无环，故 $G_1 \triangleq G[E_1] - v_0$ 与 $G_2 \triangleq G[E_2] - v_0$ 都不是空图．于是 $V_1 \triangleq V(G_1)$ 与 $V_2 \triangleq V(G_2)$ 为所求．

(3) ⇒ (4)　因为(4) 是(3) 的特例．

(4) ⇒ (1)　因为 v_0 在每一条 w-u 路上，故 w 与 v 在 $G - v_0$ 不连通，由定理 2.4，v_0 是分离点．定理证毕．

推论 2.6　设 T 是树，则 $v \in V(T)$ 是分离点的充要条件是，v 不是树叶．

证　**必要性**　分离点的度大于等于 2，故不是树叶．

充分性　设 v 与不同顶点 w, u 相邻，因为 T 是树，所以 $P = wvu$ 是唯一的 w-u 路且过 v．由定理 2.5(4)，v 是分离点．　证毕．

定义 2.3　设 $e \in E(G)$，若 $\omega(G - e) = \omega(G) + 1$，则称 e 是图 G 的桥．

例 2.3　由定义，桥就是只有一个元素的余圈．一般说，一个图可能有分离点而无桥，如图 2.3(a)；可能有桥而无分离点，如(b)；也可能既有桥又有分离点，如(c)；也可能都没有，如(d)．

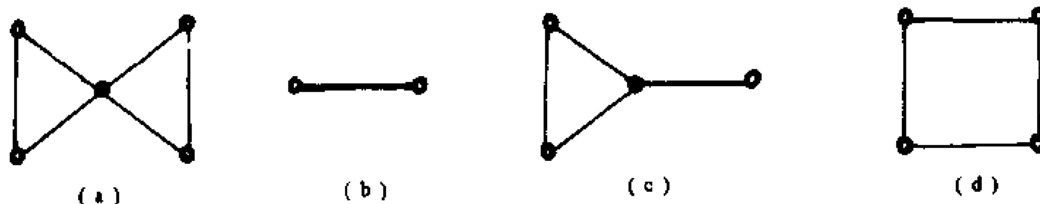


图 2.3　图的分离点(用实心点表示)和桥(用粗线表示)

定理 2.7　设 G 连通，$e \in E(G)$，下列各条等价：

(1) e 是桥；

(2) G 的每个生成树含 e；

(3) e 不在任何圈上；

(4) 存在 $V(G)$ 的一个分解 V_1, V_2，对任意 $u \in V_1, v \in V_2$，e 在每条 u-v 路上；

(5) 存在 $u, v \in V(G)$，e 在每条 u-v 路上．

证　(1) ⇒ (2)　若 G 有生成树 T 不含 e，则 $T + e$ 含圈，故 $G - e$ 连通，矛盾．

(2) ⇒ (3)　设 T 是生成树，则 $e \in E(T)$. 若 e 在某个圈上，则有圈 C 使 $\{e'\} = E(C)\backslash E(T)$. 由习题 2.9，$T' \triangleq (T-e)+e'$ 是生成树，但 T' 不含 e，矛盾.

(3) ⇒ (4)　设 $e=(x,y)$，并令

$$V_1 = \{u \in V(G) \mid 存在\ u\text{-}x\ 路不含\ e\},$$
$$V_2 = \{v \in V(G) \mid 存在\ v\text{-}y\ 路不含\ e\},$$

因为 C 不在任何圈上，所以 V_1 与 V_2 为所求分解.

(4) ⇒ (5)　因为(5)是(4)的特例.

(5) ⇒ (1)　因为 e 在每条 u-v 路上，所以 u 与 v 在 $G-e$ 不连通，故 e 是桥. 定理证毕.

下面的推论是定理的直接结果.

推论 2.8　设 G 连通，下列命题等价：

(1) G 是 2 边连通的；

(2) G 没有桥；

(3) G 的每条边都在某个圈上.

推论 2.9　T 是树，当且仅当 T 的每条边都是桥.

§2.3　块

定义 2.4　有分离点的图称为**可分图**，否则称为**不可分图**. 图 G 的极大连通不可分子图称为 G 的**块**，并用 $\sigma(G)$ 或 σ 表示 G 的块数.

由定义，环、平凡图、两个顶点的无环连通图都是块. 至少有三个顶点的块是 2 连通的. 一个连通图若没有分离点，则自身就是一个块. 若 G 有分离点 v_1，则由式(2.1)，G 是由 $G[E_1]$ 与 $G[E_2]$ 在 v_1"粘接"而成. 继续考查 $G[E_1]$ 与 $G[E_2]$，由此可分解 G 为 σ 个块，而 G 正是这 σ 个块在分离点"粘接"成的.

例 2.4　如图 2.4，图 G 分解为七个块，或者说这七个块在分离点"粘接"成 G.

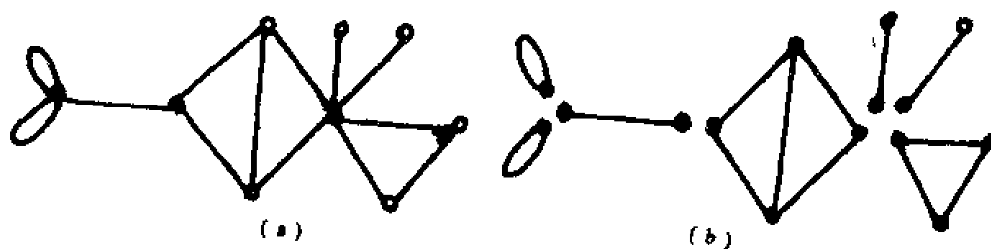


图 2.4　(a) 图 G　(b) G 分解为七个块

定理 2.10　设 G 连通，$|V(G)| \geqslant 3$，下列命题等价：

(1) G 是块；

(2) G 的任二顶点在某个圈上；

(3) G 的任一顶点和任一边在某个圈上；

(4) G 的任二边在某个圈上；

(5) G 的任二边在某个余圈上；

(6) G 的任二顶点 u 与 v，任一边 e，有 u-v 路含 e；

(7) G 的任三顶点 u,v 与 w，有 u-v 路含 w；

(8) G 的任三顶点 u,v 与 w，有 u-v 路不含 w.

证 (1) ⇒ (2) 任取顶点 $u\neq v$，令

$$X=\{x\in V(G)\mid x\text{ 在某个含 }u\text{ 的圈上，且 }x\neq u\}.$$

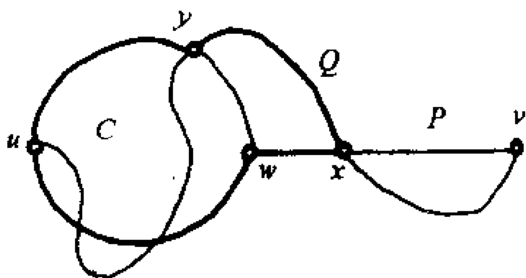


图 2.5

因为 $|V(G)|\geqslant 3$ 且 G 没有分离点，由习题 2.8，G 没有桥. 设 x 与 u 相邻，由定理 2.7，边 (x,u) 在某个圈上，故 $x\in X$，即得 $X\neq\varnothing$. 为证(2)，只需证 $v\in X$. 不然，令 $w\in X$ 且使 $d(w,v)$ 最小，设 P 是最短 w-v 路. 因为 $w\in X$，故 w 与 u 在某个圈上，设此圈为 C. 因为 w 不是分离点，由定理 2.5，有 u-v 路 Q 不过 w. 从 u 沿 Q 旅行，最后离开 C 的顶点是 y；继续沿 Q 旅行，首先到达 P 的顶点是 x，如图 2.5 所示. 于是，

$$C\text{ 的 }w\text{-}u\text{-}y\text{ 段}+Q\text{ 的 }y\text{-}x\text{ 段}+P\text{ 的 }w\text{-}x\text{ 段}$$

是含 u 与 x 的圈，在图 2.5 中用粗线标出. 由 X 的定义，$x\in X$，因为 x 在最短 w-v 路 P 上，故

$$d(x,v)<d(w,v),$$

此与 w 的选法矛盾. 故得证.

(2) ⇒ (3) 任取 $u\in V(G)$，$e=(v,w)\in E(G)$，由(2)，可设圈 C 含 u 与 v. 若 $w\in V(C)$，则 C 的 w-u-v 段加上边 (v,w) 就是含 u 与 e 的圈. 若 $w\notin V(C)$，由(2)，u 与 w 在某个圈上，故在此圈上有 u-w 路 P 不含 v. 从 u 沿 P 旅行，最后离开 C 的顶点是 y，则

$$C\text{ 的 }v\text{-}u\text{-}y\text{ 段}+P\text{ 的 }y\text{-}w\text{ 段}+e$$

是含 u 与 e 的圈.

(3) ⇒ (4) 任取 $e_1=(u_1,v_1)$，$e_2=(u_2,v_2)\in E(G)$，由(3)可设圈 C 含 v_1 与 e_2. 若 $u_1\in V(C)$，则 C 的 u_1-e_2-v_1 段加上边 e_1 就是含 e_1 与 e_2 的圈. 若 $u_1\notin V(C)$，由(3)，u_1 与 e_2 在某个圈上，故在此圈上有 u_2-u_1 路 P 不含 v_1. 从 u_2 沿 P 旅行，最后离开 C 的顶点是 y，则

$$C\text{ 的 }v_1\text{-}e_2\text{-}y\text{ 段}+P\text{ 的 }y\text{-}u_1\text{ 段}+e_1$$

是含 e_1 与 e_2 的圈.

(4) ⇒ (5) 设 (u,v)，$(w,y)\in E(G)$，由(4)，存在含此两条边的圈 C，不妨设为

$$C=uu_1\cdots u_kwyv_1\cdots v_lvu.$$

令 $S=\{u,u_1,\cdots,u_k,w\}$，则 $G[S]$ 连通. 设

$$G-S=G_1+G_2+\cdots+G_m,\ m\geqslant 1,$$

因为 $v,y\notin S$ 且连通，不妨设 $v,y\in V(G_1)$，则 $[V(G_1),\overline{V(G_1)}]$ 是含边 (u,v) 与 (w,y) 的余圈.

(5) ⇒ (6) G 的任二顶点各关联一条边，由(5)，此二边在一个余圈上. 由推论 1.12 可知此二边在一个圈上，于是我们得到(2)，从而也得到(3). 今任取顶点 u 与 v，边 e，则有圈 C_1 含 u 与 e，有圈 C_2 含 v 与 e. 若 $u\in V(C_2)$ 或 $v\in V(C_1)$，则 u,v 与 e 在同一圈上，故有 u-v 路含 e. 今设 $u\notin V(C_2)$ 且 $v\notin V(C_1)$. 从 u 沿 C_1 旅行，首先到达 C_2 的顶点是 w；再从 w 沿 C_2 或 C_2^{-1} 旅行，经 e 到达 v；由此即得 u-v 路含 e.

(6) ⇒ (7)　任三顶点 u,v 与 w,设 e 是关联于 w 的边,由(6),有 u-v 路含 e,从而也含 w.

(7) ⇒ (8)　任三顶点 u,v 与 w.由(7),有 u-w 路 P 含 v. 于是,P 的 u-v 段为所求.

(8) ⇒ (1)　设 $v \in V(G)$,再任取顶点 w 与 u,由(8),有 w-u 路不含 v. 故由定理 2.5,v 不是分离点. 所以 G 是一个块. 定理得证.

设 G 是一个图,把 G 的每个块作为一个顶点,相应顶点集设为 X;把 G 的每个分离点作为一个顶点,相应顶点集为 Y. $x \in X$ 与 $y \in Y$ 相邻,如果块 x 含分离点 y. 由此所得到的图称为图 G 的块 - 分离点图,记为 $b(G)$.

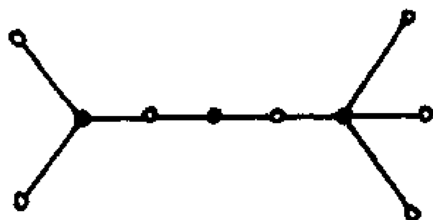


图 2.6　图 2.4(a) 的块 - 分离点图

例 2.5　图 2.4(a) 的图 G 有 3 个分离点(在这里用"•"表示);有 7 个块(在这里用"∘"表示),其块 - 分离点图如图 2.6.

定理 2.11　设 G 连通,则 G 的块 - 分离点图是树.

证　由于连通图 G 是由 G 的块在分离点"粘接"而成,所以,块 - 分离点图 $b(G)$ 是连通的. 若 $b(G)$ 有圈,由定理 1.10,必为偶圈,因此,圈上相应于块的顶点至少有两个. 此圈在 G 中对应的子图是连通的且没有分离点;由块的极大性,它的真子图不会是块,此与至少含两个块矛盾.　**证毕.**

由于连通图的块 - 分离点图是树,故又称块 - 分离点树.

推论 2.12　以 $\sigma(v)$ 表 G 的含顶点 v 的块数,则

$$\sigma = \omega + \sum_{v \in V(G)} (\sigma(v) - 1). \tag{2.3}$$

证　只需证当 G 连通即 $\omega = 1$ 时式(2.3)成立. 若 v 不是分离点,则 $\sigma(v) = 1$. 不妨设 $v_1, \cdots, v_r$ 是分离点,则 $b(G)$ 的顶点数 $= \sigma + r$,边数 $= \sum_{i=1}^{r} \sigma(v_i)$. 因为 $b(G)$ 是树,所以

$$\sum_{i=1}^{r} \sigma(v_i) = \sigma + r - 1,$$

从而式(2.3)在 $\omega = 1$ 时成立.　**证毕.**

§2.4　基本圈和基本余圈

定义 2.5　设 G 连通图,T 生成树,则

$$\bar{T} \triangleq G - E(T),$$

称为**余树**. 余树的边称为**连枝**或**弦**.

定义 2.6　设 G 连通图,T 生成树,由定理 2.1,T 加上任一连枝有唯一圈,称为**基本圈**.

定理 2.13　设 G 是(n,m)连通图,则 G 恰有$(m - n + 1)$个基本圈.

证　设 T 是 G 的生成树,则 T 有$(n - 1)$条树枝,故余树 $\bar{T}$ 有$(m - n + 1)$条连枝. 由定理 2.1,每条连枝决定一个不同的基本圈,故恰有$(m - n + 1)$个基本圈.　**证毕.**

从定理 2.13 直接导出下列推论.

推论 2.14 设 G 是 (n,m) 图，则 G 恰有 $(m-n+\omega)$ 个基本圈.

例 2.6 给定(6,8)图如图 2.7(a)，生成树 T 有 5 条树枝，余树 $\overline{T}$ 有 3 条连枝，则图 G 有 3 个基本圈.

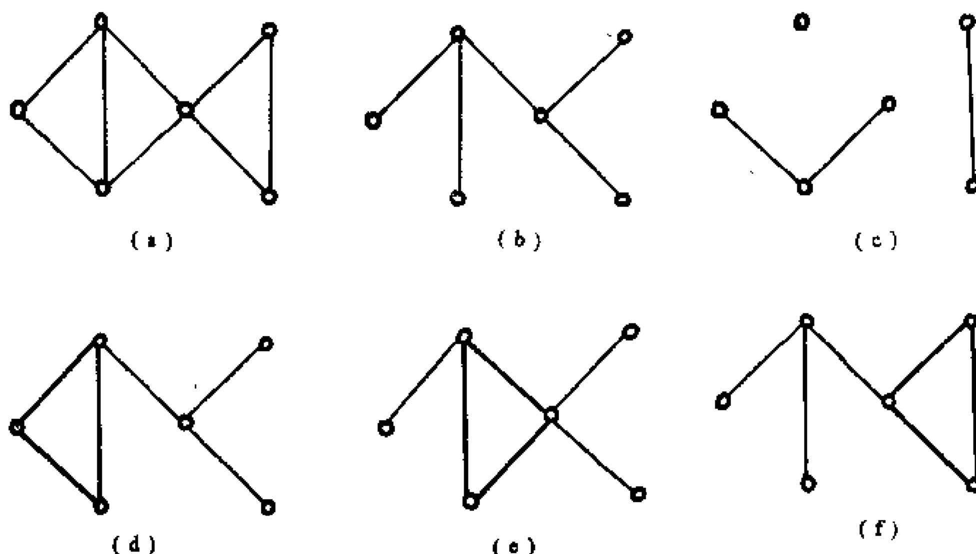


图 2.7 (a) 图 G (b) 生成树 T (c) 余树 $\overline{T}$ (d) ~ (f) 3 个基本圈(用粗线表示)

定理 2.15 设 T 是连通图 G 的生成树，则

(1) 余树 $\overline{T}$ 不含余圈，

(2) $\overline{T}+e$ 含唯一余圈，$e \in E(T)$.

证 (1) 设 F 是余圈则 $G-F$ 不连通，故 $G-F$ 不包含生成树 T，因而 F 不含在 $\overline{T}$ 中.

(2) 设 $T-e$ 的两个连通支是 T_1, T_2. 设 $S=V(T_1)$，则

$$\overline{S} \triangleq V \backslash S = V(T_2).$$

因为 T_1 是 $G[S]$ 的生成子图，故 $G[S]$ 连通，同理 $G[\overline{S}]$ 连通. 所以 $[S,\overline{S}]$ 是余圈并且包含在 $\overline{T}+e$ 中. 为证明唯一性只需证明包含在 $\overline{T}+e$ 中的任一余圈 F 都满足

$$[S,\overline{S}] \subseteq F.$$

假设有 $e_0 \in [S,\overline{S}], e_0 \notin F$，则 $e_0 \in E(\overline{T})$，故 $(T-e)+e_0$ 是生成树. 因为 $F \subseteq \overline{T}+e$，所以

$$T-e \subseteq G-F.$$

于是

$$(T-e)+e_0 \subseteq G-F,$$

故 $G-F$ 连通，此与 F 是余圈相矛盾. 证毕.

定义 2.7 设 G 连通图，$\overline{T}$ 是生成树 T 的余树，由定理 2.15，$\overline{T}$ 加上任一树枝有唯一余圈，称为**基本余圈**.

定理 2.16 设 G 是 (n,m) 连通图，则 G 恰有 $(n-1)$ 个基本余圈.

证 因为 G 的生成树有 $(n-1)$ 条树枝，每条树枝决定一个不同的基本余圈，故得证.

由定理 2.16 直接得到下面的推论.

推论 2.17 (n,m) 图恰有 $(n-\omega)$ 个基本余圈.

例 2.7 图 G,生成树及余树如图 2.7(a) ~ (c),其基本余圈有 5 个,如图 2.8 所示.

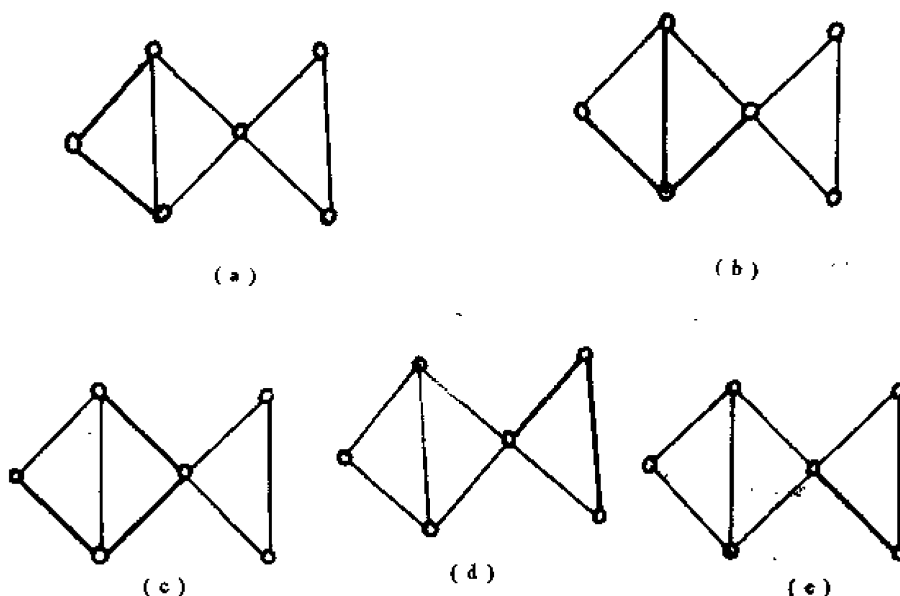


图 2.8 图 2.7(a) 的基本余圈(用粗线表示)

§2.5 有向树

定义 2.8 设 G 是有向图,若 G 的基础图是树,则称 G 是**有向树**,也简称为**树**.

设 T 是有向树.若存在顶点 v_0 可达其余每个顶点,则称 T 为**外向树**,v_0 为**始根**.若存在顶点 v_0,其余每个顶点都可达 v_0,则称 T 为**内向树**,v_0 为**终根**.

外向树和内向树统称为**有根树**.始根和终根统称为**根**.

例 2.8 如图 2.9,(a) 是树,(b) 是外向树,(c) 是内向树.(b) 与(c) 都是有根树.

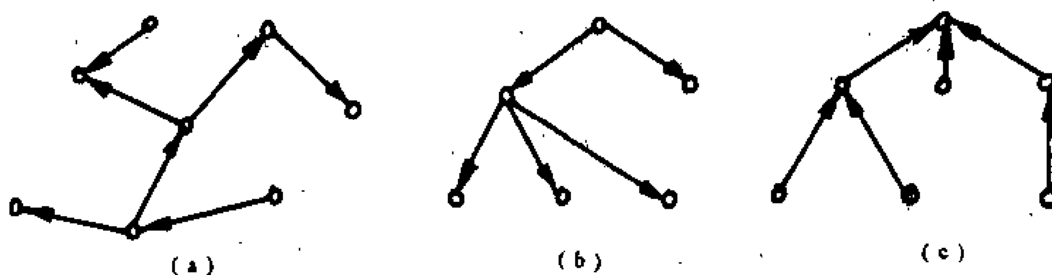


图 2.9 树和有根树

外向树正像自然界中的一棵不断生长着的树.它可表示一个组织系统的组织结构图,如军事组织系统,工会组织系统,行政组织系统,家族系统等.外向树也常称**家族树**.在家族树中,

(1) 若有边(u,v),则称 v 是 u 的**子顶点**,u 是 v 的**父顶点**;

(2) 若 v,w 都是 u 的子顶点,则称 v 与 w 是**兄弟顶点**;

(3) 若 $u \neq x$,u 可达 x,则称 u 是 x 的**前代顶点**,x 是 u 的**后代顶点**.

内向树如同细川入江的河系图.它可表示淘汰赛的比赛图,树叶表示参赛者,分枝点表示比赛获胜者,根表示获得冠军者.

外向树与内向树具有互逆性质，关于外向树的命题容易移植到内向树中，因此下面仅对外向树作进一步讨论.

定义 2.9 称有向图 G 是拟强连通的，如果对任二顶点 u,v 都有顶点 w，使 w 可达 u 与 v.

从定义直接导出强连通是拟强连通的，拟强连通是连通的.

定理 2.18 设 T 是非平凡有向图，下列命题等价：

(1) T 是外向树；

(2) 存在 $v_0 \in V(T)$，对每个顶点 $v \neq v_0$ 有唯一的 v_0-v 有向路；

(3) T 拟强连通，任删一边不再拟强连通；

(4) T 拟强连通，且存在 $v_0 \in V(T)$ 使

$$d^-(v_0) = 0,\ d^-(v) = 1 \quad \forall\ v \neq v_0; \tag{2.4}$$

(5) T 没有圈且满足式(2.4)；

(6) T 连通且满足式(2.4)；

(7) T 拟强连通且没有圈；

(8) T 拟强连通，$|E(T)| = |V(T)| - 1$.

证 (1) ⇒ (2) 外向树的根满足(2).

(2) ⇒ (3) 由(2)，T 是拟强连通的. 若删去边 $e = (u,v)$ 仍是拟强连通的，则存在顶点 w，在 $T - e$ 中有 w-u 有向路，也有 w-v 有向路. 于是在 T 中有 2 条 w-v 有向路，与(2) 中唯一性相矛盾.

(3) ⇒ (4) 由习题 2.21，存在顶点 v_0 可达其余每个顶点，故

$$d^-(v) \geqslant 1,\ v \neq v_0.$$

若存在 $v \neq v_0$，$d^-(v) > 1$，则 v 至少有两条入边，删去其中一条边，v_0 仍可达其余每个顶点，即 T 仍是拟强连通的，与(3) 矛盾. 故式(2.4) 对 $v \neq v_0$ 得证. 如果 v_0 有入边，则删去该入边后，v_0 仍可达其余每个顶点，即 T 仍是拟强连通的，与(3) 矛盾. 故式(2.4) 对 v_0 也得证.

(4) ⇒ (5) 由式(2.4)，有

$$|E(T)| = \sum_{v \in V(T)} d^-(v) = |V(T)| - 1, \tag{2.5}$$

又 T 是连通的，故 T 无圈.

(5) ⇒ (6) 由式(2.4) 导出式(2.5)，又 T 无圈，故 T 是连通的.

(6) ⇒ (7) 由式(2.4) 导出式(2.5)，又 T 连通，故 T 无圈. 今从任一顶点 $v \neq v_0$ 出发沿边的相反方向旅行，因为 $d^-(v) = 1$，故这样的旅行是唯一确定的. 又因 T 无圈故旅行中每个顶点恰好经过一次. 由于顶点数有限，旅行必终止在没有入边的 v_0. 换言之，每个顶点都从 v_0 可达，由习题 2.21，T 是拟强连通的.

(7) ⇒ (8) 因为 T 连通且无圈，故 T 是树，于是(8) 得证.

(8) ⇒ (1) 因为 T 连通且 $|E(T)| = |V(T)| - 1$，故 T 是树. 又由习题 2.21，存在 v_0 可达其余每个顶点，故 T 是外向树.

至此定理证毕.

定义 2.10 设 T 是外向树，v_0 是树根，对 T 的任一顶点 v，v_0-v 有向路的长称为顶点 v

的**层数**.特别地,树根 v_0 在零层.如果外向树的每一层的顶点都规定了次序,则称为**有序树**.

设 T 是有序树.如果把 T 的顶点分层画出来,零层在最上面,然后按层数由小到大依次从上画到下,同层顶点画在同一水平线上,并且同层顶点的次序依从左到右排列.这样画出的有序树可以省去有向边的箭头而不会发生误解.

例 2.9 如图 2.10(a) 的有序树共有 4 层,取消箭头之后可画为(b).

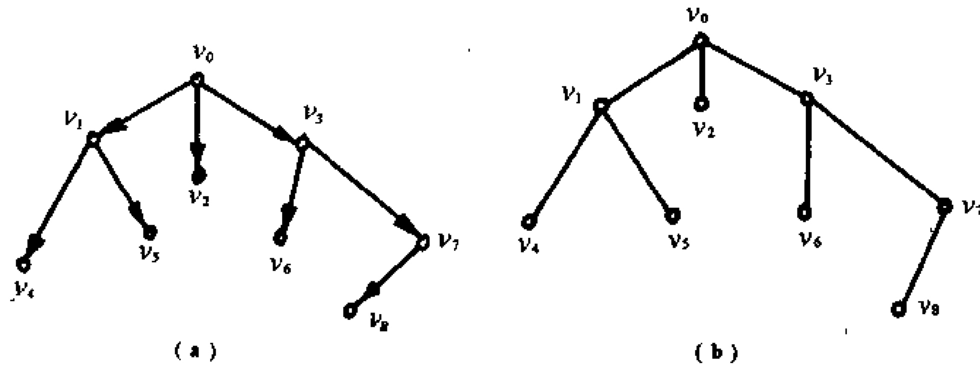


图 2.10 有序树的表示

定义 2.11 设 T 是外向树,$v \in V(T)$,令

$$S = \{u \in V(T) \mid u = v \text{ 或 } u \text{ 是 } v \text{ 的后代顶点}\},$$

则由 S 导出的子图是以 v 为根的外向树,称为 T 的**以 v 为根的子树**.

定义 2.12 设 T 是外向树.若 T 的每个分枝点至多有 m 个子顶点,则称 T 为 **m 元树**;若每个分枝点恰有 m 个子顶点,则称为 **m 元正则树**.

若 m 元正则树的所有树叶的层数相同,则称为 **m 元完全正则树**.

例 2.10 在图 2.11 中分别画出 3 元树,3 元正则树和 3 元完全正则树.

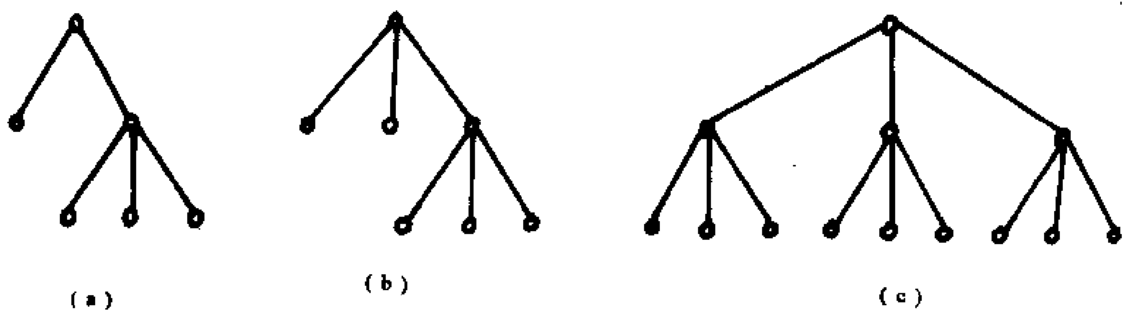


图 2.11 (a) 3 元树 (b) 3 元正则树 (c) 3 元完全正则树

定理 2.19 设 T 是 2 元正则树,r 为分枝点数,I 为各分枝点层数之和,L 为各树叶层数之和,则

$$L = I + 2r. \tag{2.6}$$

证 对分枝点数 r 归纳证明.$r = 1$ 时,$I = 0$,$L = 2$,故式(2.6)成立.设 $r = k$ 时式(2.6)成立.今设 $r = k + 1$.若树高为 h,则有树叶 v_i 其层数是 h.因为 T 是 2 元正则树,所以 v_i 有兄弟 v_j,v_j 也是 h 层的.设 v_i 与 v_j 的父顶点是 v_t,则

$$T' \triangleq T - \{v_i, v_j\}$$

是 2 元正则树而 v_t 是树 T' 的树叶，并且 T' 的分枝点数是$(r-1)$，各分枝点层数之和是$(I-(h-1))$，各树叶层数之和是 $L-2h+(h-1)=L-h-1$，由归纳假设，

$$L-h-1=I-(h-1)+2(r-1),$$

从而式(2.6)成立．　证毕．

§2.6　应用——最短路问题之一

定义 2.13　若图 G 的每一条边 e 都对应一个实数 $w(e)$，则称 $w(e)$ 为边 e 的权，并称图 G 为赋权图．图 G 的子图 H 的权记为 $w(H)$，定义为

$$w(H)\triangleq\sum\{w(e)\mid e\in E(H)\}. \tag{2.7}$$

在实际问题中，权 $w(e)$ 可表示距离，时间，容量，费用，利润，成本，效益，亏损等等．

例 2.11　如图 2.12，(a) 是赋权有向图，(b) 是赋权无向图．

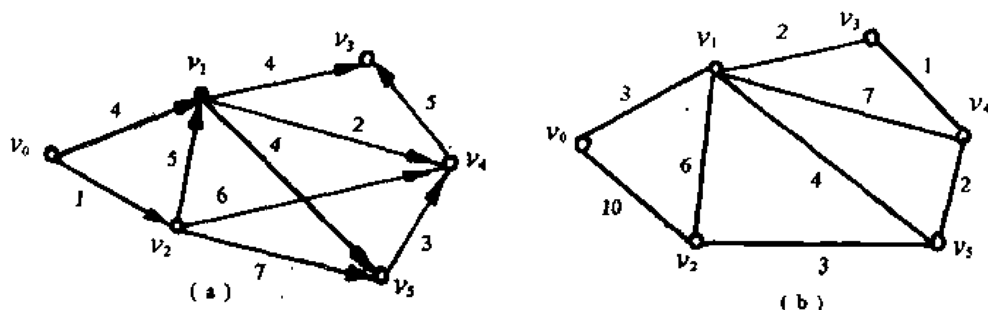


图 2.12　赋权图

设 G 是赋权图且 $w(e)\geqslant 0, e\in E(G)$．若 P 是 v_i-v_j 路，则路 P 的权 $w(P)$ 称为 P 的长．长最小的 v_i-v_j 路称为**最短路**．v_i-v_j 最短路的长称为顶点 v_i 到 v_j 的**距离**，记为 $d(v_i,v_j)$．如果没有 v_i-v_j 通道，则规定 $d(v_i,v_j)=\infty$．最短路问题之一是如何求指定顶点到其余顶点的最短路和距离．

为方便，以 i 表顶点 v_i，并设 $V(G)=\{0,1,\cdots,n-1\}$．若$(i,j)\in E(G)$ 则 $w(i,j)$ 表边 (i,j) 的权．若 $(i,j)\notin E(G)$，规定 $w(i,j)=\infty$．设 P_{0j} 表示 v_0-v_j 最短路，并记 $d_{0j}\triangleq w(P_{0j})$．

最短路有一个重要而明显的性质：最短路是一条路，且最短路的任一段也是最短路．假设在 v_0-v_j 最短路中只取一条，则对无向图，由定理 2.1，从 v_0 到其余顶点的最短路将构成一棵树；对有向图，由定理 2.18，从 v_0 到其余顶点的最短路将构成一棵以 v_0 为树根的外向树．由此得到启示，求指定顶点 v_0 到其余顶点的最短路，可采用树生长的过程．实现这一过程的方法是 E. W. Dijkstra 在 1959 年提出来的，一般称为 Dijkstra 算法．

设已经生长到树的顶点集为 S，并设 $\bar{S}=V(G)\backslash S$．在开始时，$S=\{0\}$，并给图的顶点如下标号：

$$l(0)=0,\ l(j)=w(0,j),\ j\in\bar{S}.$$

这样的标号自然满足两个条件：

$$l(i)=d_{0i},\ i\in S. \tag{2.8}$$

$$l(j)=\min\{l(i)+w(i,j)\mid i\in S\},\ j\in\bar{S}. \tag{2.9}$$

如何继续树生长过程?我们有下面的引理.

引理 2.20 设 S 为已经生长到树的顶点集,且顶点的标号满足式(2.8)与(2.9).令

$$l(k) = \min \{l(j) \mid j \in \overline{S}\} < \infty, \tag{2.10}$$

则 $l(k) = d_{0k}$,且在 P_{0k} 上只有 $k \in \overline{S}$,其余顶点都在 S 中.

证 由式(2.9)不妨设 $l(k) = l(i) + w(i,k)$,则 $l(k)$ 是一条 v_0-v_k 路的长.于是

$$l(k) \geqslant d_{0k}. \tag{2.11}$$

今从 v_0 沿最短 v_0-v_k 路 P_{0k} 旅行,首先到达 $\overline{S}$ 的顶点是 v_j,并设 P_1 是 P_{0k} 的 v_0-v_j 段,则由式(2.9)与式(2.10),有

$$d_{0k} \geqslant w(P_1) \geqslant l(j) \geqslant l(k). \tag{2.12}$$

由式(2.11)与式(2.12)引理得证.

由引理 2.20,若将 k 生长到 S 中,则式(2.8)满足.为了满足式(2.9),对 $j \in \overline{S}\backslash\{k\}$,令

$$l(j) = \min \{l(j), l(k) + w(k,j)\}. \tag{2.13}$$

事实上,把式(2.9)代入式(2.13),有

$$\begin{aligned} l(j) &= \min \{\min \{l(i) + w(i,j) \mid i \in S\},\ l(k) + w(k,j)\} \\ &= \min \{l(i) + w(i,j) \mid i \in S \cup \{k\}\}, \end{aligned}$$

至此已解决了求 v_0 到其余顶点的距离的问题.如何求出最短路呢?若最短 v_0-v_i 路是

$$P_{0i} = v_0 \cdots v_j v_i,$$

则称 v_j 是 v_i 的紧前顶点.因此只要在树生长过程中记录生长到 S 的顶点的紧前顶点,则可逆向追踪求出最短路.我们用 $t(i) = j$ 表示顶点 v_i 的紧前顶点是 v_j.在开始时,顶点 v_i 的紧前顶点都是 v_0,即

$$t(i) = 0,\ i \neq 0,$$

当顶点 k 生长到 S 后,若顶点 j 的标号不变,则其紧前顶点也不变;若顶点 j 由式(2.13)得到新标号

$$l(j) = l(k) + w(k,j),$$

则改变顶点 j 的紧前顶点为顶点 k,即令 $t(j) = k$.

以上分析得到求最短路和距离的算法.

算法 2.1 Dijkstra 算法——求某一顶点到其余顶点的最短路

设 G 是赋权图且 $w(e) \geqslant 0, e \in E(G)$,$S$ 表生长到树中的顶点集,$\overline{S} = V(G)\backslash S$.

第 1 步 令 $S = \{0\}, l(0) = 0$,

$$l(j) = w(0,j),\ t(j) = 0,\ j \in \overline{S}.$$

第 2 步 令

$$l(k) = \min \{l(j) \mid j \in \overline{S}\}, \tag{2.14}$$

若 $l(k) = \infty$,逆向追踪,停.否则令

$$S = S \cup \{k\}.$$

第 3 步 若 $\overline{S} = \varnothing$,逆向追踪,停.否则,对每个 $j \in \overline{S}$,若 $l(j) > l(k) + w(k,j)$ 则令

$$l(j) = l(k) + w(k,j), \tag{2.15}$$

$$t(j) = k, \tag{2.16}$$

转第 2 步.

算法 2.1 对无向图和有向图都是适用的.

例 2.12　对图 2.12(a) 的有向图 G,求顶点 v_0 到其余顶点的最短路,并画出所得外向树.

算法 2.1 的计算过程可用下表表出,表中打 * 者表示按式(2.14) 得出的 k.

迭代 \ $l(i),t(i)$ \ 顶点	0	1	2	3	4	5	当前生长点
1	0	4,0	*1,0	∞,0	∞,0	∞,0	0
2		*4,0		∞,0	7,2	8,2	2
3				8,1	*6,1	8,2	1
4				*8,1		8,2	4
5						*8,2	3
6	逆向追踪						5

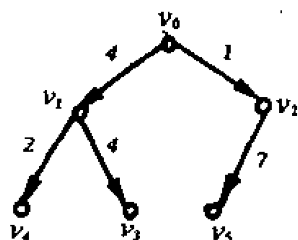


图 2.13　以 v_0 为根的外向树

逆向追踪: $t(5) = 2,\ t(2) = 0,$

$t(4) = 1,\ t(1) = 0,\ t(3) = 1,$

于是得以 v_0 为根的外向树,如图 2.13.

§2.7　应用——最短路问题之二

最短路的另一问题是求任意两顶点间的最短路. 设 G 是赋权图,权 $w(e)$ 是实数. 在这种情形,可能出现有 u-v 通道而没有最短 u-v 路,即在 u-v 通道中包含了权为负的闭通道,称为负回路. 若有 u-v 通道包含负回路,则规定 $d(u,v) = -\infty$. 于是在权 $w(e)$ 为实数的情形,顶点 u 与 v 的距离可如下定义:

$$d(u,v) = \begin{cases} +\infty, & \text{没有 } u\text{-}v \text{ 通道}, \\ w(P), & P \text{ 是最短 } u\text{-}v \text{ 路}, \\ -\infty, & \text{否则}. \end{cases} \tag{2.17}$$

求任意两顶点间的最短路的算法是下面要讨论的 Floyd 算法.

设 $G = (V,E)$ 是赋权图,$w(e)$ 是实数,$|V| = n$,并以 P_{ij} 表最短 v_i-v_j 路,以 d_{ij} 表 v_i 到 v_j 的距离. 设 $V = \{v_1,v_2,\cdots,v_n\}$.

首先假定 G 没有负回路,欲求任意两顶点间的距离. 为方便,设

$$V_{ij}^k \triangleq \{v_i,v_j\} \cup \{v_1,\cdots,v_k\},\quad G_{ij}^k \triangleq G[V_{ij}^k],\quad E_{ij}^k \triangleq E(G_{ij}^k),$$

$$V_{ii}^k \triangleq \{v_i\} \cup \{v_1,\cdots,v_k\},\quad G_{ii}^k \triangleq G[V_{ii}^k],\quad E_{ii}^k = E(G_{ii}^k),$$

并用 P_{ij}^k, d_{ij}^k 分别表示在 G_{ij}^k 中顶点 v_i 到 v_j 的最短路和距离. 在开始时,令

$$\begin{cases} d_{ii}^0 = 0, \\ d_{ij}^0 = \begin{cases} w(v_i, v_j), (v_i, v_j) \in E \\ +\infty \quad , 否则 \end{cases} \quad (i \neq j). \end{cases} \tag{2.18}$$

我们希望从 d_{ij}^{k-1} 推出 d_{ij}^k,从而当 $k = n$ 时得 $d_{ij} = d_{ij}^n$. 事实上,在 G_{ij}^k 中的 v_i-v_j 通道有两种情形:

(1) 所有 v_i-v_j 通道都不过 v_k,这时 $d_{ij}^k = d_{ij}^{k-1}$;

(2) 存在 v_i-v_j 通道过 v_k,这时

$$d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}. \tag{2.19}$$

显然,情形(1) 包含在情形(2) 之中,故式(2.19) 可作为从 d_{ij}^{k-1} 到 d_{ij}^k 的递推公式. 于是,从式(2.18) 开始,用式(2.19) 递推,当 $k = n$ 时即得

$$(d_{ij})_{n\times n} = (d_{ij}^n)_{n\times n}.$$

其次,如何判断 G 是否有负回路?当 G 没有负回路时必有 $d_{ii}^k = 0, k = 0,1,\cdots,n$. 如果

$$d_{ii}^0 = d_{ii}^1 = \cdots = d_{ii}^{k-1} = 0,\ d_{ii}^k < 0,$$

则由式(2.19) 导出 $d_{ik}^{k-1} + d_{ki}^{k-1} < 0$,即在 G_{ii}^k 中有 v_i 到 v_i 的负回路.

最后是如何找出最短路或负回路的问题. 只要能记录 P_{ij}^k 的终点 v_j 的紧前顶点就能逆向追踪找出最短路或负回路. 为方便,若顶点 v_j 在 P_{ij}^k 上的紧前顶点是 v_l,则记 $\theta_{ij}^k = l$. 在开始时可设

$$\theta_{ij}^0 = i,\ i,j = 1,2,\cdots,n.$$

继续迭代有两种可能:

(1) $d_{ij}^{k-1} \leqslant d_{ik}^{k-1} + d_{kj}^{k-1}$,这时,由式(2.19),$d_{ij}^k = d_{ij}^{k-1}$ 故

$$\theta_{ij}^k = \theta_{ij}^{k-1}.$$

(2) $d_{ij}^{k-1} > d_{ik}^{k-1} + d_{kj}^{k-1}$,这时,由式(2.19),$d_{ij}^k = d_{ik}^{k-1} + d_{kj}^{k-1}$,故

$$\theta_{ij}^k = \theta_{kj}^{k-1}.$$

至此,当权为实数时,求任意两顶点间的最短路问题完全得到解决.

算法 2.2 Floyd 算法——求任意两顶点间的最短路

设 G 是赋权图,权为实数,$|V(G)| = n$.

第 1 步　按式(2.18) 计算 d_{ij}^0,并令 $\theta_{ij}^0 = i$,$(i,j = 1,2,\cdots,n)$.

令 $k = 1$.

第 2 步　按式(2.19) 计算 d_{ij}^k,并令

$$\theta_{ij}^k = \begin{cases} \theta_{ij}^{k-1},\ 当\ d_{ij}^{k-1} \leqslant d_{ik}^{k-1} + d_{kj}^{k-1}, \\ \theta_{kj}^{k-1},\ 否则. \end{cases} \quad i,j = 1,2,\cdots,n \tag{2.20}$$

第 3 步　若存在 $d_{ii}^k < 0, 1 \leqslant i \leqslant n$,则从 v_i 逆向追踪到 v_i 求出负回路,停.

第 4 步　若 $k < n$,令 $k = k + 1$,转第 2 步.

第 5 步　得任意两顶点间的距离 $d_{ij} = d_{ij}^n$,且当 $d_{ij} < +\infty$ 时,逆向追踪求出最短路

$$P_{ij}, (i,j = 1,2,\cdots,n),$$

停.

例 2.13　如图 2.14 的有向图，求任意两顶点间的最短路和距离.

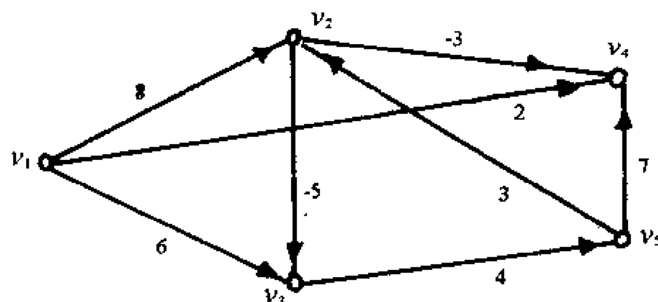


图 2.14　赋权有向图

在迭代中，用矩阵形式(d_{ij}^k)与(θ_{ij}^k)，而 $d_{ik}^{k-1}+d_{kj}^{k-1}$ 用表格形式是方便的. d_{ij}^k 右上角打"*"表示 $d_{ij}^k=d_{ik}^{k-1}+d_{kj}^{k-1}$，$\theta_{ij}^k$ 右上角打"*"表示 $\theta_{ij}^k=\theta_{kj}^{k-1}$.

$k=0$

$$(d_{ij}^0)=\begin{pmatrix}0 & 8 & 6 & 2 & \infty\\ \infty & 0 & -5 & -3 & \infty\\ \infty & \infty & 0 & \infty & 4\\ \infty & \infty & \infty & 0 & \infty\\ \infty & 3 & \infty & 7 & 0\end{pmatrix},\qquad (\theta_{ij}^0)=\begin{pmatrix}1 & 1 & 1 & 1 & 1\\ 2 & 2 & 2 & 2 & 2\\ 3 & 3 & 3 & 3 & 3\\ 4 & 4 & 4 & 4 & 4\\ 5 & 5 & 5 & 5 & 5\end{pmatrix}.$$

$k=1$

1行 / 1列	0	8	6	2	∞
0	0	8	6	2	∞
∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞

，　$(d_{ij}^1)=(d_{ij}^0)$，$(\theta_{ij}^1)=(\theta_{ij}^0)$.

$k=2$

2行 / 2列	∞	0	−5	−3	∞
8	∞	8	3	5	∞
0	∞	0	−5	−3	∞
∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞
3	∞	3	−2	0	∞

，

$$(d_{ij}^2)=\begin{pmatrix}0 & 8 & 3^* & 2 & \infty\\ \infty & 0 & -5 & -3 & \infty\\ \infty & \infty & 0 & \infty & 4\\ \infty & \infty & \infty & 0 & \infty\\ \infty & 3 & -2^* & 0^* & 0\end{pmatrix},$$

$$(\theta_{ij}^2)=\begin{pmatrix}1 & 1 & 2^* & 1 & 1\\ 2 & 2 & 2 & 2 & 2\\ 3 & 3 & 3 & 3 & 3\\ 4 & 4 & 4 & 4 & 4\\ 5 & 5 & 2^* & 2^* & 5\end{pmatrix}.$$

$k=3$

3 行 \ 3 列	∞	∞	0	∞	4
3	∞	∞	3	∞	7
-5	∞	∞	-5	∞	-1
0	∞	∞	0	∞	4
∞	∞	∞	∞	∞	∞
-2	∞	∞	-2	∞	2

,

$$(d_{ij}^{3})=\begin{pmatrix}0 & 8 & 3 & 2 & 7^{*}\\ \infty & 0 & -5 & -3 & -1^{*}\\ \infty & \infty & 0 & \infty & 4\\ \infty & \infty & \infty & 0 & \infty\\ \infty & 3 & -2 & 0 & 0\end{pmatrix},$$

$$(\theta_{ij}^{3})=\begin{pmatrix}1 & 1 & 2 & 1 & 3^{*}\\ 2 & 2 & 2 & 2 & 3^{*}\\ 3 & 3 & 3 & 3 & 3\\ 4 & 4 & 4 & 4 & 4\\ 5 & 5 & 2 & 2 & 5\end{pmatrix}.$$

$k=4$

4 行 \ 4 列	∞	∞	∞	0	∞
2	∞	∞	∞	2	∞
-3	∞	∞	∞	-3	∞
∞	∞	∞	∞	∞	∞
0	∞	∞	∞	0	∞
0	∞	∞	∞	0	∞

, $(d_{ij}^{4})=(d_{ij}^{3})$, $(\theta_{ij}^{4})=(\theta_{ij}^{3})$.

$k=5$

5 行 \ 5 列	∞	3	-2	0	0
7	∞	10	5	7	7
-1	∞	2	-3	-1	-1
4	∞	7	2	4	4
∞	∞	∞	∞	∞	∞
0	∞	3	-2	0	0

,

$$(d_{ij}^{5})=\begin{pmatrix}0 & 8 & 3 & 2 & 7\\ \infty & 0 & -5 & -3 & -1\\ \infty & 7^{*} & 0 & 4^{*} & 4\\ \infty & \infty & \infty & 0 & \infty\\ \infty & 3 & -2 & 0 & 0\end{pmatrix},$$

$$(\theta_{ij}^{5})=\begin{pmatrix}1 & 1 & 2 & 1 & 3\\ 2 & 2 & 2 & 2 & 3\\ 3 & 5^{*} & 3 & 2^{*} & 3\\ 4 & 4 & 4 & 4 & 4\\ 5 & 5 & 2 & 2 & 5\end{pmatrix}.$$

于是$(d_{ij})=(d_{ij}^{5})$，$(\theta_{ij})=(\theta_{ij}^{5})$. 当 $d_{ij}<\infty$ 时逆向追踪分别得出 5 棵外向树，如图 2.15.

以 v_1 为根： $\theta_{15}=3$，$\theta_{13}=2$，$\theta_{12}=1$，$\theta_{14}=1$.

以 v_2 为根： $\theta_{25}=3$，$\theta_{23}=2$，$\theta_{24}=2$.

以 v_3 为根： $\theta_{35}=3$，$\theta_{34}=2$，$\theta_{32}=5$.

以 v_4 为根： $\theta_{41}=\theta_{42}=\theta_{43}=\theta_{45}=4$.

以 v_5 为根： $\theta_{54}=2$，$\theta_{52}=5$，$\theta_{53}=2$.

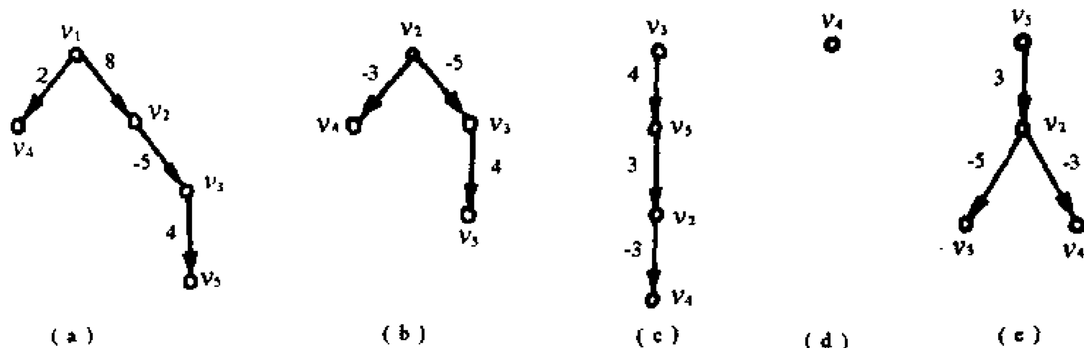


图 2.15

如上所述,Floyd 算法能判断是否有负回路,且当有负回路时求出负回路,当没有负回路时求出任意两顶点间的最短路和距离.但应注意,无向图没有负回路的充要条件是每条边的权非负.故对无向图总假定权 $w(e) \geqslant 0, e \in E(G)$.

§2.8 应用——最小生成树

例 2.14 有 n 个城市,要修建通讯网络连接这些城市,已知城市 v_i 与 v_j 间建立直接通讯系统的成本为 c_{ij},如何建造总成本最少的通讯网络?

以顶点代表城市,在两个城市建立直接通讯系统的成本作为相应两个顶点的边的权,则建造总成本最少的通讯网络的问题就是求赋权连通图的权最小的生成树,称为**最小生成树**.

由定理 2.1,连通图 G 的极大无圈子图是 G 的生成树.换言之,我们可以用逐步加边的方法得到 G 的一个子图 T,使其具有性质:

(1) T 不含圈;

(2) 任给 T 加一边都含圈,

这样的 T 就是一棵生成树.J. B. Jr. Kruskal 在 1956 年把上述加边的方法推广,提出在加边时总是加余下的所有边中权最小的边,即在不含圈的条件下尽可能选权最小的边,从而构成权最小的极大无圈子图——最小生成树.

为了选取权尽可能小的边,可事先对 G 的边排序,即令 $E(G)=\{e_1,e_2,\cdots,e_m\}$ 满足

$$w(e_1) \leqslant w(e_2) \leqslant \cdots \leqslant w(e_m). \tag{2.21}$$

为了判断 T 是否含圈,可给 T 的所有顶点加上 l 标号满足

$$l(u)=l(v) \Longleftrightarrow u \text{ 与 } v \text{ 在 } T \text{ 连通}. \tag{2.22}$$

于是,当考查边 $e_k=(u,v)$ 时,由式(2.22),有

$$T+e_k \text{ 含圈} \Longleftrightarrow l(u)=l(v).$$

至此得最小生成树的算法.

算法 2.3 Kruskal 算法——求最小生成树

设 G 是赋权连通 (n,m) 图,G 的边的排序满足式(2.21),$V(G)=\{v_1,\cdots,v_n\}$.

第 1 步　令 $T = V(G)$.

$$l(v_j) = j,\ j = 1,\cdots,n,$$
$$k = 1, i = 0.$$

第 2 步　设 $e_k = (u,v)$，若 $l(u) = l(v)$ 转第 5 步. 否则令

$$T = T + e_k,\ i = i + 1.$$

第 3 步　对一切满足 $l(v_j) = \max\{l(u), l(v)\}$ 的 v_j，令

$$l(v_j) = \min\{l(u), l(v)\}.$$

第 4 步　若 $i = n - 1$，T 是最小生成树，停.

第 5 步　令 $k = k + 1$，转第 2 步.

例 2.15　如图 2.16(a) 的图 G，求最小生成树. 首先给 G 的边按式(2.21)排序，如(b). 然后给顶点标号，如(c). 从(d) 到(g) 依次加 4 条边得最小生成树.

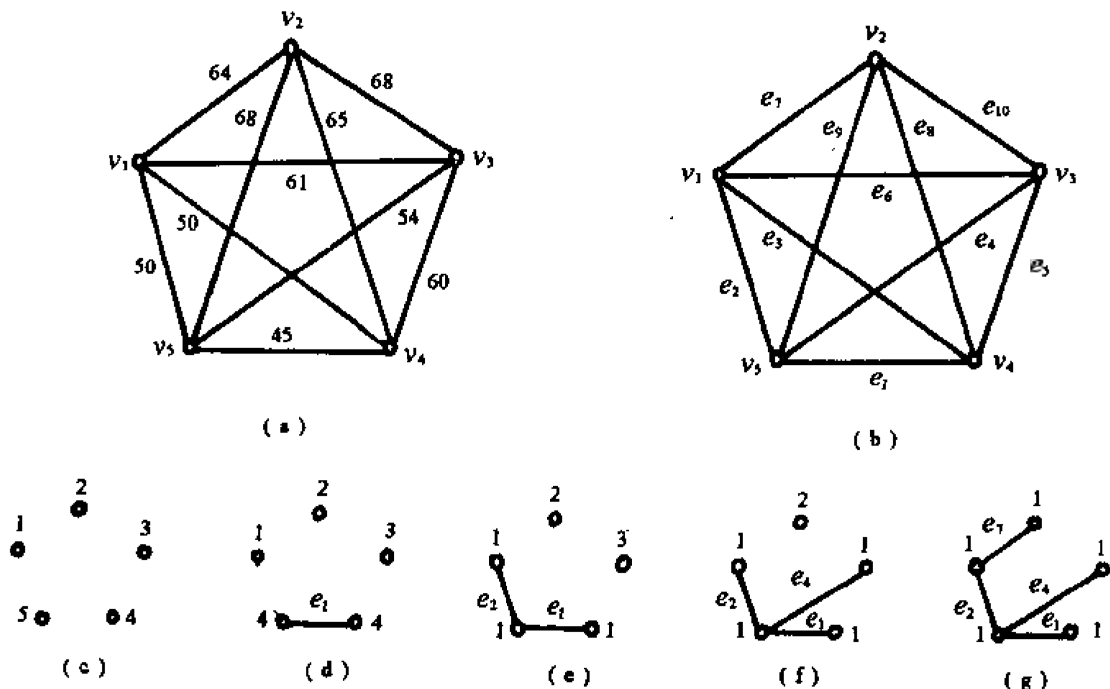


图 2.16　求最小生成树的过程

在本节的最后我们证明，由 Kruskal 算法求出最小生成树.

引理 2.21　由 Kruskal 算法得出的生成树是最小生成树.

证　假设由 Kruskal 算法得出的生成树 T 不是最小生成树，不妨设

$$T = [e_1, e_2, \cdots, e_{n-1}].$$

对异于 T 的任意生成树 T_1，令

$$f(T_1) = \min\{i \mid e_i \notin E(T_1),\ 1 \leqslant i \leqslant n - 1\},$$

今设 T_1 是一棵最小生成树且使 $f(T_1)$ 尽可能大. 不妨设 $f(T_1) = k$，于是

$$e_1, \cdots, e_{k-1} \in E(T) \cap E(T_1),$$
$$e_k \in E(T) \backslash E(T_1),$$

由定理 2.1，$T_1 + e_k$ 有唯一圈 C. 设

$$e_k' \in (E(C) \cap E(T_1)) \backslash E(T),$$

则 $T_1' \triangleq (T_1 + e_k) - e_k'$ 是生成树. 另一方面，T_1 的子图

$$[e_1, \cdots, e_{k-1}, e_k']$$

是无圈的，由算法，$w(e_k') \geqslant w(e_k)$，故

$$w(T_1') = w(T) + w(e_k) - w(e_k') \leqslant w(T_1)$$

从而 T_1' 也是最小生成树. 但 $f(T_1') > k = f(T_1)$，与 T_1 的选法相矛盾.　**证毕.**

§2.9　应用——最优 2 元树

在计算机和通讯中常用二进制编码表示字符. 例如，用 1 表示 A，用 00 表示 B，用 001 表示 C，用 100 表示 D. 但当我们接收到信息 001100 时，却无法分辨这是 CD，或 BAD，或 CAB，还是 $BAAB$？为了解决这个问题，可用前缀码.

定义 2.14　设 $a_1a_2\cdots a_{n-1}a_n$ 是长为 n 的符号串，称其子串 $a_1, a_1a_2, \cdots, a_1a_2\cdots a_{n-1}$ 分别是长为 $1, 2, \cdots, n-1$ 的**前缀**. 设 $A = \{\beta_1, \beta_2, \cdots, \beta_m\}$ 是 m 个符号串的集合，若对任意的 $\beta_i, \beta_j \in A, i \neq j$，$\beta_i$ 与 β_j 互不为前缀，则称 A 为**前缀码**. 若符号串 β_i 只出现 0 与 1 两个符号，$i = 1, \cdots, m$，则称 A 为**二元前缀码**.

例 2.16　本节开始时所作编码

$$\{1, 00, 001, 100\}$$

不是二元前缀码. 而编码

$$\{1, 01, 001, 000\} \tag{2.23}$$

是二元前缀码.

引理 2.22　一棵 2 元树产生一个二元前缀码，并且由 2 元正则树产生的二元前缀码是唯一的.

证　设 T 是 2 元树，有 t 片树叶. 对 T 的每个分枝点，若有两个子顶点，则到左子顶点的边标 0，到右子顶点的边标 1；若只有一个子顶点，则可任意作为左子顶点其边标 0，或作为右子顶点其边标 1. 设 v_i 是任一片树叶，把从树根到树叶 v_i 的唯一路 P_i 各边的标号组成的符号串放在 v_i 处；这个符号串的前缀都在这条唯一路 P_i 上. 因此，2 元树的 t 片树叶由此产生的 t 个符号串构成一个二元前缀码.

如果 2 元树是正则的，则每个分枝点恰有两个子顶点，故所产生的二元前缀码是唯一的.　**证毕.**

图 2.17

例 2.17　如图 2.17 的 2 元正则树产生的唯一前缀码是

$$A = \{1, 01, 000, 001\}.$$

这正是式(2.23)给出的.

一般地，在信息传输中，各符号出现的频率是不相同的. 当已知传输符号出现的频率时，如何选择前缀码，使传输的二进制位尽可能少？这个问题归结为产生一个最优 2 元树.

定义 2.15 设 T 为 2 元树，若 t 片树叶 v_i 分别对应实数 w_i，则称 w_i 是树叶 v_i 的**权**，$i=1,\cdots,t$，而 $w(T)=\sum_{i=1}^{t}w_i l_i$ 称为 T 的**权**，其中 l_i 为树叶 v_i 的层数，$i=1,\cdots,t$，并称 T 为**带权 2 元树**. 在所有带权 $w_1,\cdots,w_t$ 的 t 片树叶的 2 元树中，权最小的 2 元树称为**最优 2 元树**.

引理 2.23 设 T 是带权为 $w_1\leqslant w_2\leqslant\cdots\leqslant w_t$ 的最优 2 元树，则一定能使带权为 w_1 与 w_2 的顶点是兄弟，且它们的层数都等于树高.

证 设树叶 v_i 的层数最高，若 $i>1$，则交换树叶 v_i 与 v_1 的权，得到新的带权 2 元树 T'，且 $w(T')\leqslant w(T)$. 因为 T 是最优 2 元树，故 T' 也是最优 2 元树，且带权 w_1 的顶点 v_i 的层数等于树高. 如果 v_i 没有兄弟，则去掉 v_i 并令 v_i 的父顶点带权 w_1，由此得到一个权更小的 2 元树，矛盾. 因此 v_i 必有兄弟顶点，设为 v_j. 类似上面的证明可得，v_j 是除去 w_1 外权最小的顶点，即 v_j 是带权 w_2 的顶点. 证毕.

引理 2.24 设有一棵带权 $w_1+w_2,w_3,\cdots,w_t$ 的最优 2 元树 T'，其中 $w_1\leqslant w_2\leqslant\cdots\leqslant w_t$，让带权 w_1+w_2 的顶点产生两个子顶点分别带权 w_1 和 w_2，则所得到的 T 是带权 w_1，$w_2,\cdots,w_t$ 的最优 2 元树.

证 由条件，

$$w(T)=w(T')+w_1+w_2. \tag{2.24}$$

设 T_1 是带权 $w_1,\cdots,w_t$ 的最优 2 元树，由引理 2.23，带权 w_1 与 w_2 的顶点 v_1 与 v_2 是兄弟，且它们的层数是树高. 设它们的父顶点是 v，令 $T_1'=T_1-\{v_1,v_2\}$ 并让 v 带权 w_1+w_2，则

$$w(T_1)=w(T_1')+w_1+w_2. \tag{2.25}$$

因为 T_1 是最优的，故 $w(T_1)\leqslant w(T)$. 若 $w(T_1)<w(T)$，则由式(2.24)与式(2.25)，$w(T_1')<w(T')$，此与 T' 是最优的相矛盾. 从而 $w(T_1)=w(T)$. 证毕.

由引理 2.24 得出求带权 $w_1\leqslant w_2\leqslant\cdots\leqslant w_t$ 的最优 2 元树的方法，称为 Huffman 算法.

算法 2.4 Huffman 算法——求最优 2 元树

设权 $w_1,w_2,\cdots,w_t$ 是实数，令

$$S=\{w_1,w_2,\cdots,w_t\}.$$

第 1 步 令 $w_s=\min\{w_i|w_i\in S\}$，

$w_r=\min\{w_i|w_i\in S\backslash\{w_s\}\}$.

第 2 步 作带权 w_s 的顶点 v_s 在左，带权 w_r 的顶点 v_r 在右，并作 v_s 与 v_r 的父顶点 v 带权 w_s+w_r.

第 3 步 令 $S=(S\backslash\{w_s,w_r\})\cup\{w_s+w_r\}$.

第 4 步 若 $|S|=1$，得带权 $w_1,w_2,\cdots,w_t$ 的最优 2 元树，停. 否则转第 1 步.

例 2.18 求带权 2,4,5,7,8,11 的最优 2 元树.

由算法 2.4，对给定的权 $w_1\leqslant\cdots\leqslant w_t$，最高层带权 w_1 与 w_2，依次从最高层到零层，S 中的元素个数递减 1，故所得最优 2 元树共 $(t-1)$ 层. 这里有六个权，故经 5 次迭代，得共有 5 层的一棵最优 2 元树 T(图 2.18(e))，其权

$$w(T)=7\times2+8\times2+5\times3+2\times4+4\times4+11\times2=91.$$

算法过程由图 2.18(a) ~ (e) 给出.

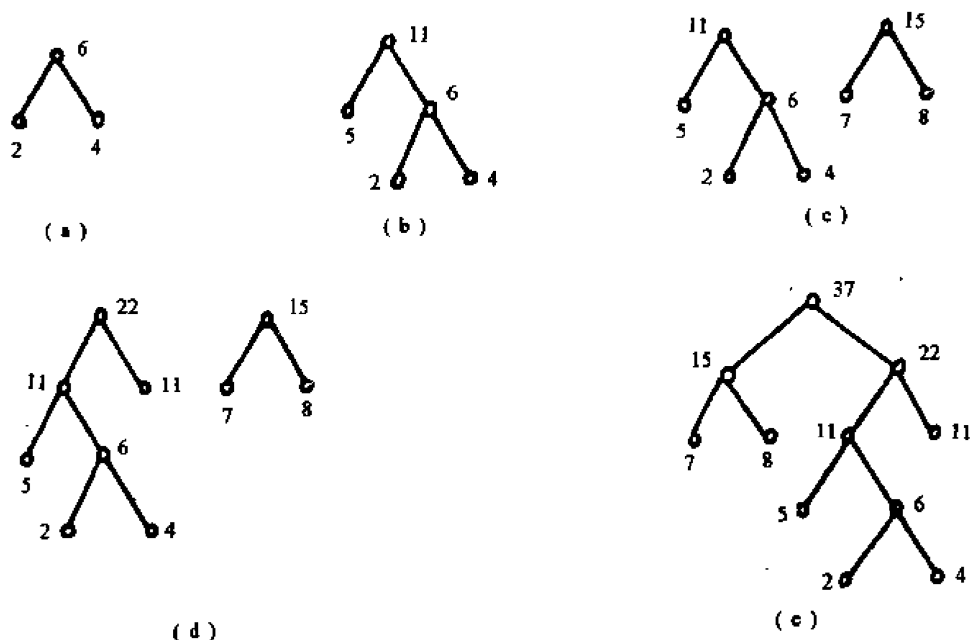


图 2.18

例 2.19 已知字母 A,B,C,D,E,F 出现的频率：

$A-35\%$， $B-10\%$， $C-20\%$， $D-10\%$， $E-20\%$， $F-5\%$，

求一个前缀码使传输的二进制位最少.

作带权 35,10,20,10,20,5 的最优 2 元树 T，得一前缀码. 若树叶 v_i 带权 w_i，则 v_i 处的符号串表示出现频率为 $w_i\%$ 的字母，如下表.

字母	A	B	C	D	E	F
符号串	11	010	00	0111	10	0110

传输 100 个这样的字母所用二进制位为

$$4\times(5+10)+3\times10+2\times(20+20+35)=240.$$

最优 2 元树 T 及树叶处的符号串如图 2.19.

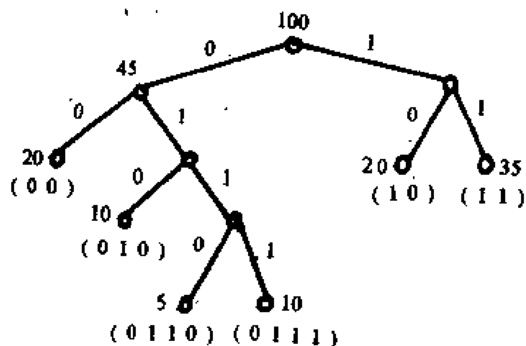


图 2.19

习　题　二

2.1　用至少两个另外的方法证明推论 2.2.

2.2　仅有二片树叶的树是一条路.

2.3　T 是树，$\Delta \geqslant k$，则 T 至少有 k 片树叶.

2.4　T 是树，$u_0 \in V(T)$ 满足

$$\max_{v \in V(T)} d(u_0, v) = \min_{u \in V(T)} \{\max_{v \in V(T)} d(u, v)\},$$

则称 u_0 是 T 的中心，证明

(1) 当 $|V(T)| \geqslant 3$，树叶不是中心，

(2) 只有一个中心或两个相邻中心.

2.5　G 连通，则 $|E(G)| \geqslant |V(G)| - 1$.

2.6　非平凡连通图至少有两个非分离点.

2.7　设 G 是非平凡简单连通图，则 G 是一条路当且仅当 G 恰有两个非分离点.

2.8　当 $|V(G)| \geqslant 3$，若有桥则必有分离点.

2.9　设 T 是连通图 G 的生成树，C 是 G 的圈，$e \in E(T) \cap E(C)$，$\{e'\} = E(C) \backslash E(T)$，则

$$T' \triangleq (T - e) + e'$$

是 G 的生成树.

2.10　设 G 有 ω 个连通支，则下列命题等价：

(1) G 是森林；

(2) $|E(G)| = |V(G)| - \omega$；

(3) G 的每条边都是桥.

2.11　若图 G 有桥，则 G 必有奇顶点.

2.12　证明图 G 的两个块至多有一个公共顶点.

2.13　$v \in V(G)$ 是分离点的充要条件是，v 至少是 G 的两个块的顶点.

2.14　G 是 2 边连通的，当且仅当 G 的任意两个顶点都有两条边不交的路.

2.15　设 G 简单连通图，则 G 没有桥的充要条件是，存在 G 的定向图是强连通的.

2.16　设 H 是连通图 G 的块 - 分离点树，则 H 的任意两片树叶间的距离都是偶数.

2.17　设 G 连通，$|V(G)| \geqslant 3$，则下列命题等价：

(1) G 没有桥；

(2) G 的任二顶点在某个闭迹上；

(3) G 的任一顶点和任一边在某个闭迹上；

(4) G 的任二边在某个闭迹上；

(5) G 的任二顶点 u,v 和任一边 e，都有 u-v 迹含 e；

(6) G 的任二顶点 u,v 和任一边 e，都有 u-v 迹不含 e；

(7) G 的任三顶点 u,v,w，都有 u-v 迹含 w.

2.18　设 G 连通，则 G 的余圈必含树枝.

2.19　图的任一圈和任一余圈，若有公共边则必为偶数条公共边.

2.20　如图 2.20，T 是 G 的生成树，求 G 的所有基本圈和基本余圈.

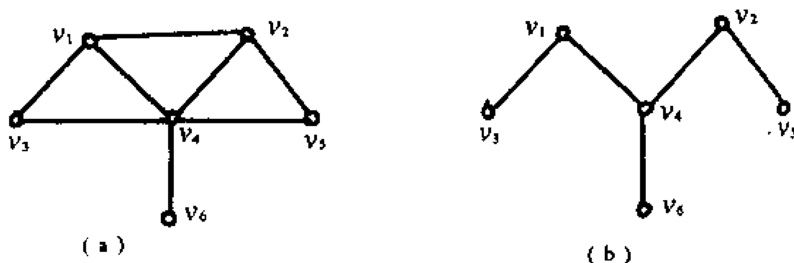


图 2.20　(a) 图 G　(b) 生成树 T

2.21　设 G 有向图，则 G 是拟强连通的充要条件是存在 $v_0 \in V(G)$，v_0 可达其余每个顶点.

2.22　设 G 有向图，则 G 含外向树为生成树的充要条件是 G 是拟强连通的.

2.23　设 T 是 m 元正则树，其树叶数为 t，分枝点数为 r，则

$$(m-1)r = t-1.$$

2.24　设 T 是二元正则树，其树叶数为 t，则

$$E(T) = 2t-2.$$

2.25　设 $E' \subseteq E(G)$，则 E' 含余圈的充要条件是

$$\omega(G-E') > \omega(G).$$

2.26　设 $E' \subseteq E(G)$，则 E' 含余圈的充要条件是 G 的每个生成森林都含 E' 的边.

2.27　G 连通，则 G 的任一圈必含连枝.

2.28　$E' \subseteq E(G)$，则 E' 含圈的充要条件是 G 的每个余森林含 E' 的边.

2.29　设 T 是连通 (n,m) 图 G 的生成树，C_i 是对应连枝 e_i 的基本圈，则 $e_{i_1},\cdots,e_{i_r}$ 都在

$$C_{i_1} \oplus C_{i_2} \oplus \cdots \oplus C_{i_r}$$

中，$1 \leqslant r \leqslant m-n+1$.

2.30　图 G 的相异二个闭迹的对称差或为一圈，或为边不交的圈之并.

2.31　连通 (n,m) 图 G 的任意闭迹或为基本圈，或为基本圈的对称差.

2.32　连通 (n,m) 图 G 的闭迹个数 e_0 满足

$$m-n+1 \leqslant e_0 \leqslant 2^{m-n+1}-1.$$

2.33　利用习题 2.31，试给出找连通图的所有圈的步骤，并由此步骤求图 2.20(a) 的图 G 的所有圈.

2.34　图的相异二个边割的对称差或是一余圈，或是边不交余圈之并.

2.35　设 $\bar{T}$ 是连通 (n,m) 图 G 的余树，F_i 是对应树枝 e_i 的基本余圈，则 $e_{i_1},\cdots,e_{i_r}$ 都在

$$F_{i_1} \oplus F_{i_2} \oplus \cdots \oplus F_{i_r}$$

中，$1 \leqslant r \leqslant n-1$.

2.36　连通 (n,m) 图 G 的任意边割或为基本余圈，或为基本余圈的对称差.

2.37　连通(n,m)图G的边割数ν_0满足

$$n-1 \leqslant \nu_0 \leqslant 2^{n-1}-1.$$

2.38　利用习题2.35,试给出找连通图的所有余圈的步骤,并由此步骤求图2.20(a)的图G的所有余圈.

2.39　对图2.12(b)的无向图G,用Dijkstra算法求顶点v_0到其余顶点的最短路和距离,并画出由最短路构成的生成树.

2.40　对图2.21的有向图G,用Dijkstra算法求顶点v_0到其余顶点的最短路和距离,并画出由最短路构成的外向树.

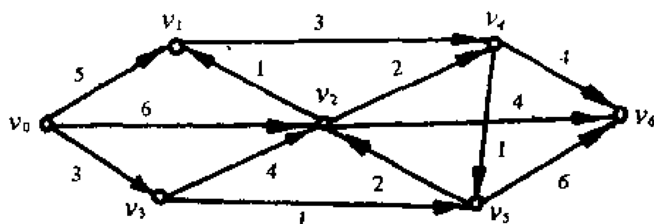


图2.21

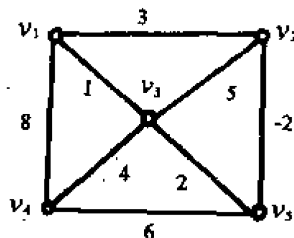


图2.22

2.41　如图2.22的无向图G,用Floyd算法判断有无负回路,若有则求出负回路.

2.42　如图2.23的有向图G,用Floyd算法判断有无负回路,若有则求出负回路.

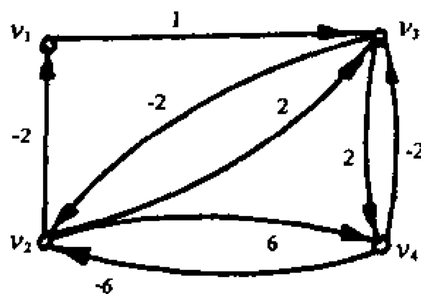


图2.23

2.43　如图2.12(b)的无向图,用Floyd算法求任意两顶点间的距离,并分别画出从每个顶点到其余顶点的最短路所构成的生成树.

2.44　如图2.12(a)的有向图,用Floyd算法求任意两顶点间的距离,并分别画出从每个顶点到其余顶点的最短路所构成的外向树.

2.45　设G是赋权连通图,$e_0 \in E(G)$不是环,且$w(e_0) \leqslant w(e)$, $e \in E(G)$,则存在G的一棵最小生成树含e.

2.46　设(n,m)图G是连通的,则由定理2.1,G的极大无圈子图是G的生成树.试由此设计一个求生成树的算法,并用此算法求图2.24的图的生成树.

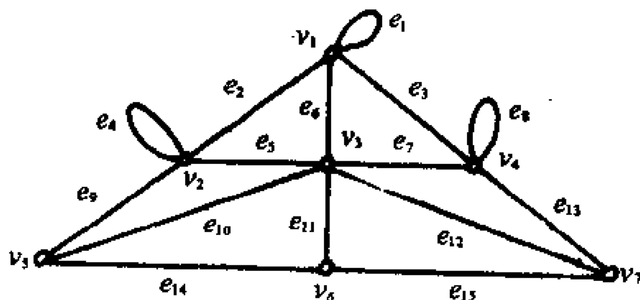


图2.24

2.47 设(n,m)图G连通，则由定理2.1，G的极小连通子图是G的生成树. 试由此设计一个求生成树的算法，并由此算法求图2.24的图的生成树.

2.48 连通图G的生成树T_1与T_2的距离$d(T_1,T_2)$定义为

$$d(T_1,T_2) \triangleq |E(T_1)\backslash E(T_2)|.$$

试证此距离满足距离公理：

(1) $d(T_i,T_j) \geqslant 0$ 并且 $d(T_i,T_j) = 0$ 当且仅当 $T_i = T_j$，

(2) $d(T_i,T_j) = d(T_j,T_i)$，

(3) $d(T_i,T_j) \leqslant d(T_i,T_k) + d(T_k,T_j)$.

2.49 设G连通图，$T_1,\cdots,T_r$是G的所有生成树，生成树的距离由习题2.48定义. 今以G的生成树为顶点，并且

$$T_i \text{与} T_j \text{相邻} \iff d(T_i,T_j) = 1,$$

由此得到的图称为连通图G的**树图**. 试证连通图的树图是连通的.

2.50 用Kruskal算法求图2.22的图的最小生成树.

2.51 赋权连通图G的所有生成树中权最大的生成树称为**最大生成树**. 试修改Kruskal算法以得出求连通图G的最大生成树的算法，并用此算法求图2.22的图的最大生成树.

2.52 Kruskal算法是用"加边"求最小生成树，也可用"删边"的方法. 设G是赋权连通图，每次任取一圈，并删去圈上权最大的边，直到删边后的图不含圈终止.

(1) 证明算法终止时得到最小生成树，

(2) 给出计算步骤，

(3) 由(2)的算法求图2.22的图的最小生成树.

2.53 设G是赋权连通图G的生成树，则T是最小生成树的充要条件是，没有与T的距离为1而权更小的生成树.

2.54 设G是赋权连通图G的生成树，则T是最小生成树的充要条件是，对每条树枝$e \in E(T)$，余树$\overline{T}$加e的唯一余圈F满足

$$w(e) \leqslant w(e'),\ e' \in E(F).$$

2.55 设G是赋权连通图G的生成树，则T是最小生成树的充要条件是，对每条连枝$e \in E(\overline{T})$，$T + e$的唯一圈C满足

$$w(e) \geqslant w(e'),\ e' \in E(C).$$

2.56 作一棵带权1.5，2.1，0.8，3.3，6.4，5.1与4.4的最优二元树，并计算其权.

2.57 设有6个符号在通讯中出现的频率是$A - 0.25$，$B - 0.28$，$C - 0.15$，$D - 0.12$，$E - 0.14$，$F - 0.06$，求最佳前缀码，问传输1000个符号需多少二进制位.

2.58 若规定用树叶表示参加运算的元素，分枝点表示相应的运算，则利用二元有序正则树可以表达二元运算的算式，试用此法分别表示下列算式：

(1) $((((a + (b \times c)) \times d) - e) \div (f + g)) + ((h \times i) \times ((j \div k) - l))$，

(2) $(((a - b) \times c) \div ((d + e) \div f)) - ((g + h) - ((i \times j) \div k))$.

2.59 对于二元有序正则树有下列行遍方法(在这里，行遍即每个顶点访问一次而周游一棵树)：

(1) **中序行遍法**,其访问顺序是左子树,树根,右子树,

(2) **前序行遍法**,其访问顺序是树根,左子树,右子树,

(3) **后序行遍法**,其访问顺序是左子树,右子树,树根.

利用上述三种行遍法,一个四则运算的算式相应有三种表示法,分别称为**中缀符号法**,**前缀符号法**,**后缀符号法**.中缀之意为运算符作用于紧前紧后二个元,前缀之意为运算符作用于紧后二个元,后缀之意为运算符作用于紧前二个元.因此,对中缀符号法,可按运算符的优先次序省去部分括号,对前缀和后缀符号法,可省去全部括号,所得运算式称为简化式.

试用上述三种方法分别表示习题 2.58 中的两个算式,然后再写出简化式.

第三章　圈空间和余圈空间及其应用

§ 3.1　闭迹向量和边割向量

定义 3.1　设无向图 G 的边集 $E=\{e_1,\cdots,e_m\}$，C 是一个闭迹，则 $\mu=(\mu_1,\cdots,\mu_m)^T$ 称为**闭迹向量**，其中

$$\mu_j=\begin{cases}1, & e_j\in E(C),\\ 0, & \text{否则}.\end{cases}\tag{3.1}$$

若 G 是有向图，任给 C 指定一个方向，并令

$$\mu_j=\begin{cases}1, & e_j\in E(C^+),\\ -1, & e_j\in E(C^-),\\ 0, & \text{否则},\end{cases}\tag{3.2}$$

其中　　$C^+=\{e_j\in E(C)\,|\,e_j$ 的方向与 C 的方向一致$\}$，

　　　　$C^-=\{e_j\in E(C)\,|\,e_j$ 的方向与 C 的方向相反$\}$.

若 C 上的边是 $e_{i_1},\cdots,e_{i_t}$，则可把 C 视为边导出子图而记为 $C=[e_{i_1},\cdots,e_{i_t}]$.

定义 3.2　设无向图 G 的边集 $E=\{e_1,\cdots,e_m\}$，$N(S)$ 是边割，则 $\nu=(\nu_1,\cdots,\nu_m)^T$ 称为**边割向量**，其中

$$\nu_j=\begin{cases}1, & e_j\in N(S),\\ 0, & \text{否则}.\end{cases}\tag{3.3}$$

若 G 是有向图，则

$$\nu_j=\begin{cases}1, & e_j\in N^+(S),\\ -1, & e_j\in N^-(S),\\ 0, & \text{否则},\end{cases}\tag{3.4}$$

其中　　$N^+(S)=\{e_j\in N(S)\,|\,e_j$ 的尾在 $S\}$，

　　　　$N^-(S)=\{e_j\in N(S)\,|\,e_j$ 的头在 $S\}$.

例 3.1　如图 3.1(a) 是无向图，(b) 是有向图，其闭迹向量与边割向量如表 3.1 与表 3.2.

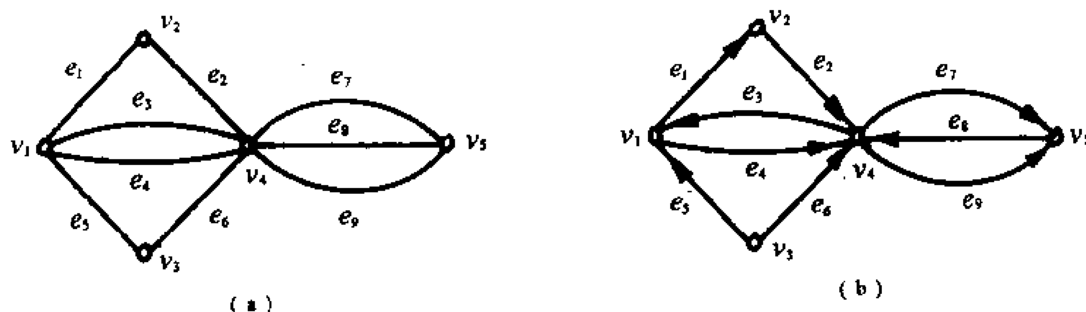


图 3.1

表 3.1 图 3.1(a) 的闭迹向量与边割向量

闭 迹	闭迹向量
$[e_1,e_2,e_4]$ $[e_1,e_2,e_4,e_7,e_9]$	$(1,1,0,1,0,0,0,0,0)^T$ $(1,1,0,1,0,0,1,0,1)^T$
边割 $N(S)$	**边割向量**
$S=\{v_5\}$ $S=\{v_1,v_4\}$	$(0,0,0,0,0,0,1,1,1)^T$ $(1,1,0,0,1,1,1,1,1)^T$

表 3.2 图 3.1(b) 的闭迹向量与边割向量

闭 迹	闭迹向量
$[e_1,e_2,e_4]$ $[e_1,e_2,e_4,e_7,e_9]$	$(1,1,0,-1,0,0,0,0,0)^T$ $(1,1,0,-1,0,0,-1,0,1)^T$
边割 $N(S)$	**边割向量**
$S=\{v_5\}$ $S=\{v_1,v_4\}$	$(0,0,0,0,0,0,-1,1,-1)^T$ $(1,-1,0,0,-1,-1,1,-1,1)^T$

定义 3.3 设图 G 是连通 (n,m) 图，有 ε_1 个圈，称为**圈数**，则以所有圈向量为行向量的 $\varepsilon_1\times m$ 矩阵称为**圈矩阵**，记为 $B(G)$ 或 B. 以所有基本圈向量为行向量的 $(m-n+1)\times m$ 矩阵称为**基本圈矩阵**，记为 B_f，它是 B 的子矩阵.

在习题 2.33 中指出，利用对称差可以从所有基本圈求所有圈. 换言之，我们可以利用模 2 加法从基本圈矩阵求圈矩阵.

算法 3.1 求圈矩阵的算法

设 G 是 (n,m) 连通图.

第 1 步 求 G 的生成树 T.

第 2 步 求出 G 的基本圈矩阵 B_f.

第 3 步 对 B_f 的行作所有可能的对称差即模 2 加法，所产生的新行连同 B_f 一起构成新矩阵 B_1.

第 4 步 在 B_1 中消去冗余行即表示边不交的圈之并的那些行.

第 5 步 B_1 中留下的行构成圈矩阵.

例 3.2 给出图 G 如图 3.2(a)，求圈矩阵.

首先求出生成树如图 3.2(b). 由此得到三个基本圈 $C_1=[e_1,e_2,e_3]$，$C_2=[e_4,e_5,e_7]$，

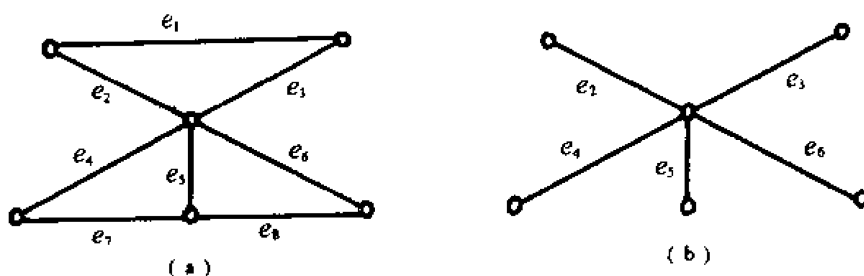


图 3.2

$C_3=[e_5,e_6,e_8]$,故

$$
B_f=\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ C_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ C_2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ C_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{matrix},
$$

$$
B_1=\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & \\ C_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & C_1 \\ C_2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & C_2 \\ C_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & C_3 \\ C_1\oplus C_2 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \text{冗余行.} \\ C_1\oplus C_3 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & \text{冗余行} \\ C_2\oplus C_3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & C_4 \\ C_1\oplus C_2\oplus C_3 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & \text{冗余行} \end{matrix}
$$

判断冗余行的方法是,当某行的边集包含另一行的边集为子集时,即当某行的元素1包含另一行的全部元素1时,该行就是冗余行,否则得到一个新的圈.如$C_1\oplus C_2$包含C_1行和C_2行,$C_1\oplus C_2\oplus C_3$包含C_1行和C_4行,故都是冗余行.当从B_1消去冗余行时,即得圈矩阵B,

$$
B=\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ C_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ C_2 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ C_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ C_4 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{matrix}.
$$

定义 3.4 设G是连通(n,m)图,有ν_1个圈,称为**余圈数**,则以所有余圈向量为行向量的$\nu_1\times m$矩阵称为**余圈矩阵**,记为$Q(G)$或Q.以所有基本余圈向量为行向量的$(n-1)\times m$矩阵称为**基本余圈矩阵**,记为Q_f,它是Q的子矩阵.

在习题2.38中指出,利用对称差可以从所有基本余圈求所有余圈.换言之,我们可以利用模2加法从基本余圈矩阵求余圈矩阵.

算法 3.2 求余圈矩阵的算法

设G是(n,m)连通图.

第1步 求G的生成树T.

第 2 步　求 G 的基本余圈矩阵 Q_f.

第 3 步　对 Q_f 的行作所有可能的对称差即模 2 加法，所产生的新行连同 Q_f 一起构成新矩阵 Q_1.

第 4 步　在 Q_1 中消去冗余行即表示边不交的余圈之并的那些行.

第 5 步　Q_1 中留下的行构成余圈矩阵.

例 3.3　给出图 G 如图 3.3(a)，求余圈矩阵.

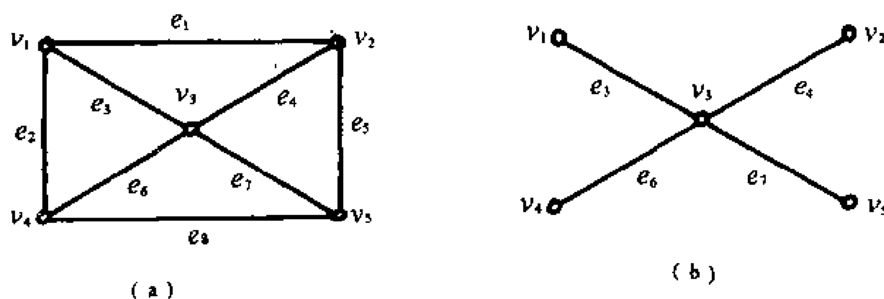


图 3.3

首先求出生成树如图 3.3(b). 由此得到四个基本余圈 $F_1=[e_1,e_2,e_3]$，$F_2=[e_1,e_4,e_5]$，$F_3=[e_2,e_6,e_8]$，$F_4=[e_5,e_7,e_8]$，故

$$Q_f=\begin{array}{c}\\ F_1\\ F_2\\ F_3\\ F_4\end{array}\begin{array}{c}\begin{array}{cccccccc}e_1&e_2&e_3&e_4&e_5&e_6&e_7&e_8\end{array}\\ \begin{pmatrix}1&1&1&0&0&0&0&0\\1&0&0&1&1&0&0&0\\0&1&0&0&0&1&0&1\\0&0&0&0&1&0&1&1\end{pmatrix}\end{array},$$

$$Q_1=\begin{array}{r|cccccccc|l}
 & e_1&e_2&e_3&e_4&e_5&e_6&e_7&e_8 & \\
F_1 & 1&1&1&0&0&0&0&0 & F_1\\
F_2 & 1&0&0&1&1&0&0&0 & F_2\\
F_3 & 0&1&0&0&0&1&0&1 & F_3\\
F_4 & 0&0&0&0&1&0&1&1 & F_4\\
F_1\oplus F_2 & 0&1&1&1&1&0&0&0 & F_5\\
F_1\oplus F_3 & 1&0&1&0&0&1&0&1 & F_6\\
F_1\oplus F_4 & 1&1&1&0&1&0&1&1 & \text{冗余行(含 }F_1,F_4)\\
F_2\oplus F_3 & 1&1&0&1&1&1&0&1 & \text{冗余行(含 }F_2,F_3).\\
F_2\oplus F_4 & 1&0&0&1&0&0&1&1 & F_7\\
F_3\oplus F_4 & 0&1&0&0&1&1&1&0 & F_8\\
F_1\oplus F_2\oplus F_3 & 0&0&1&1&1&1&0&1 & F_9\\
F_1\oplus F_2\oplus F_4 & 0&1&1&1&0&0&1&1 & F_{10}\\
F_1\oplus F_3\oplus F_4 & 1&0&1&0&1&1&1&0 & F_{11}\\
F_2\oplus F_3\oplus F_4 & 1&1&0&1&0&1&1&0 & F_{12}\\
F_1\oplus F_2\oplus F_3\oplus F_4 & 0&0&1&1&0&1&1&0 & F_{13}
\end{array}$$

从 Q_1 中消去冗余行即得余圈矩阵 Q,

$$
Q = \begin{array}{c} \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \end{array}
\begin{array}{c}
\begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0
\end{pmatrix}
\end{array}.
$$

§3.2 圈基和余圈基

定义 3.5 闭迹向量的极大无关组称为**圈基**，圈基所含向量个数称为**圈秩**，用 $\eta(G)$ 或 η 表示. 边割向量的极大无关组称为**余圈基**，余圈基所含向量个数称为**余圈秩**，用 $\xi(G)$ 或 ξ 表示.

定理 3.1 设 G 是 (n,m) 连通图，则

$$\eta = m - n + 1, \tag{3.5}$$

$$\xi = n - 1. \tag{3.6}$$

证 图 G 的基本圈有 $(m - n + 1)$ 个，每一个恰含一条弦，因此基本圈矩阵相应于这些弦的子矩阵是 $(m - n + 1)$ 阶单位阵，故基本圈向量是独立的. 又由习题 2.31，闭迹向量可由基本圈向量表示，故基本圈向量是闭迹向量的极大无关组，从而式(3.5)得证.

图 G 的基本余圈有 $(n - 1)$ 个，每一个恰含一条树枝，因此基本余圈矩阵相应于这些树枝的子矩阵是 $(n - 1)$ 阶单位矩阵，故基本余圈向量是独立的. 又由习题 2.36，边割向量可由基本余圈向量表示，故基本余圈向量是边割向量的极大无关组，从而式(3.6)得证. 定理证毕.

从定理 3.1 直接得到下面两个推论.

推论 3.2 设 (n,m) 图 G 有 ω 个连通支，则

$$\eta = m - n + \omega, \tag{3.7}$$

$$\xi = n - \omega. \tag{3.8}$$

推论 3.3 设连通 (n,m) 图 G 的圈矩阵是 B，余圈矩阵是 Q，则

$$r(B) = m - n + 1, \tag{3.9}$$

$$r(Q) = n - 1. \tag{3.10}$$

定理 3.4 强连通图可由有向圈构成圈基.

证 设 (n,m) 图 G 是强连通的，需证图 G 有 $(m - n + 1)$ 个独立有向圈.

当 $n \leqslant 2$ 时，定理是明显的. 设顶点数小于 $n(n > 2)$ 定理成立，欲证顶点数为 n 定理成立.

设 $C = [e_1, e_2, \cdots, e_k]$ 是长大于 1 的有向闭通道中长最小的，则由此直接推出：

(1) C 是有向圈，

(2) C 上任不连续两个顶点不相邻，

(3) C 上连续两个顶点可能有平行边.

今作图 G'：去掉边 $e_1, e_2, \cdots, e_k$，并用一个顶点 x' 代替 C 上 k 个顶点，不在 C 中且与 C 关联的边，以相同的边与 x' 相关联. 因为 G 强连通，所以 G' 强连通，且

$$n' \triangleq |V(G')| = n - k + 1,$$

$$m' \triangleq |E(G')| = m - k.$$

因为 $k > 1$，所以 $n' < n$. 由归纳假设，G' 有

$$\eta(G') = m' - n' + 1 = m - n$$

个独立有向圈. 相应地得出 G 有 $(m - n)$ 个独立有向圈，再加上 C，构成 G 的 $(m - n + 1)$ 个独立有向圈. **证毕.**

定理 3.5 若有向图 G 没有有向圈，则有有向余圈构成的余圈基.

证 设 G 是 (n,m) 连通有向图且没有有向圈，需证 G 有 $(n-1)$ 个独立有向余圈.

当 $n \leqslant 2$ 时定理是明显的. 设顶点数为 $n - 1(n > 2)$ 时定理成立，欲证顶点数为 n 定理成立.

若任一顶点都有长大于 1 的有向路，则

$$d^+(v) \geqslant 1, \ v \in V(G),$$

从而 G 有有向圈，矛盾. 故存在顶点 y，从 y 出发的最长有向路的长为 1，即存在顶点 x 使

(1) 有 y 到 x 的边，没有 x 到 y 的边.

(2) $d^+(x) = 0$. (3.11)

今作图 G'：去掉 y 到 x 的所有的边，并用一个顶点 x' 代替顶点 y 与 x；G 的其余的边若与 x 或 y 关联，则以相同的边与 x' 关联. 容易证明 G' 没有有向圈. 不然，设 G' 有有向圈 C'. 因为 G 不含有向圈，所以 C' 必含顶点 x'. 又 G' 在 x' 没有环，故 C' 至少含两条边. 换言之，在 C' 上，x' 有出边 e_1，也有入边 e_2. 于是，由式(3.11)，在图 G，e_1 是 y 的出边. 因为 G 没有有向圈，所以在图 G，e 是 x 的入边. 由此，从 y 沿 e_1，e_2 到 x 是长大于 1 的路，与 y 的选取矛盾.

以上证明了 G' 没有有向圈，且 $|V(G')| = n - 1$，由归纳假设，G' 有 $(n-2)$ 个独立有向余圈. 并相应得出 G 的 $(n-2)$ 个独立有向余圈，再加上

$$[\{x\}, \{\bar{x}\}]$$

这个有向余圈，构成 G 的 $(n-1)$ 个独立有向余圈. **证毕.**

§3.3 环流和势差

定义 3.6 设 G 是 (n,m) 连通有向图，$E = \{1, 2, \cdots, m\}$，若

$$\varphi = (\varphi_1, \varphi_2, \cdots, \varphi_m)^T \in R^m$$

满足

$$\sum_{i \in X^+(v)} \varphi_i = \sum_{i \in X^-(v)} \varphi_i, v \in V, \tag{3.12}$$

则称 φ 是一个**环流**，每个 φ_i 称为 $i \in E$ 的**流量**，并称式(3.12)为**守恒律**. 当 $\varphi_i \geq 0$，表示流量的方向与边 i 的方向一致；当 $\varphi_i < 0$，表示流量的方向与边 i 的方向相反.

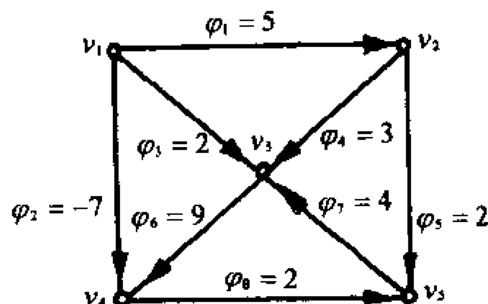


图 3.4　环流 φ

例 3.4　在图 3.4 中给出 $E = \{1,2,\cdots,8\}$，$\varphi = (5,-7,2,3,2,9,4,2)^T$，经验证在每个顶点都满足守恒律式(3.12)，故 φ 是一个环流.

定理 3.6　设 $S \subseteq V$，φ 是 G 的环流，则

$$\sum_{i \in X^+(S)} \varphi_i = \sum_{i \in X^-(S)} (\varphi_i). \tag{3.13}$$

证　由式(3.12)，

$$\sum_{v \in S} \sum_{i \in X^+(v)} \varphi_i = \sum_{v \in S} \sum_{i \in X^-(v)} \varphi_i. \tag{3.14}$$

若边 i 的两个端点都在 S 中，则边 i 是一个端点的出边，另一个端点的入边，故式(3.14)中左右两边相消，从而式(3.13)得证.　证毕.

定理 3.7　设 G 是 (n,m) 连通有向图，A 是关联矩阵，则 $\varphi \in R^m$ 是环流的充要条件是

$$A\varphi = 0. \tag{3.15}$$

证　对每个 $v_i \in V(G)$ 有

$$A_{i\cdot}\varphi = \sum_{j \in X^+(v_i)} \varphi_j - \sum_{j \in X^-(v_i)} \varphi_j,$$

于是，

φ 是环流 $\Longleftrightarrow$ 式(3.12)成立

$\Longleftrightarrow A_{i\cdot}\varphi = 0,\ v_i \in V(G)$

$\Longleftrightarrow$ 式(3.15)成立.

证毕.

例 3.5　由例 3.4 的有向图 G 和环流 φ，验证定理 3.7.

事实上，由例 3.4 有

$$A_\varphi = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \end{pmatrix} \end{array} \begin{pmatrix} 5 \\ -7 \\ 2 \\ 3 \\ 2 \\ 9 \\ 4 \\ 2 \end{pmatrix} = 0.$$

定理 3.8　有向图的闭迹向量是环流.

证　设 A 是有向图 G 的关联矩阵，C 是闭迹，则对每个 $v_i \in V(C)$，v_i 恰好与 C 的两条边关联，且有如下两种情形.

(1) 两条边对 v_i 一是出边，一是入边. 此时 $A_{i\cdot}$ 的相应分量一个"$+1$"，一个"-1"，而 C 相应的闭迹向量 μ 的分量是两个"$+1$"或两个"-1"，于是 $A_{i\cdot}\mu = 0$.

(2) 两条边对 v_i 都是出边或入边. 此时 $A_{i\cdot}$ 的相应分量是两个"$+1$"或两个"-1"，而 C 相应的闭迹向量 μ 的分量是一个"$+1$"，一个"-1"，于是 $A_{i\cdot}\mu = 0$.

以上证明 $A_{i\cdot}\mu = 0, v_i \in V(C)$，因为 μ 相应于 $E(G)\backslash E(C)$ 的分量都是零，故 $A_{i\cdot}\mu = 0, v_i \in V(G)$，所以 $A\mu = 0$，即 μ 是环流. 证毕

定理 3.9 设 G 是 (n,m) 连通有向图，则 G 的每个环流能表为圈基的线性组合.

证 设 φ 是 G 的一个环流，T 是 G 的生成树. 适当选择基本圈的方向，使基本圈向量 $\mu^1,\cdots,\mu^\nu$ 相应于弦的分量构成单位矩阵. 考虑环流 φ'，

$$\varphi' \triangleq \varphi - \sum_{i\in E\backslash E(T)} \varphi_i \mu^i, \tag{3.16}$$

则 $\varphi'_i = 0, i \in E\backslash E(T)$. 于是 $\varphi'|_T$ 是 T 上的环流，由习题 3.5，$\varphi'|_T = 0$，从而 $\varphi' = 0$. 再由式(3.16)有

$$\varphi = \sum_{i\in E\backslash E(T)} \varphi_i \mu^i.$$

故得证.

由定理 3.9 直接得出

推论 3.10 环流由余树唯一确定.

定义 3.7 设 $G = (V,E)$ 是 (n,m) 连通有向图，$V = \{1,2,\cdots,n\}$，则

$$\pi = (\pi_1,\cdots,\pi_n)^T,$$

称为 G 的**位势**，π_i 称为顶点 i 的**位势**.

设 $e = (i,j)$，令

$$\theta_e = \pi_j - \pi_i, \tag{3.17}$$

则 θ_e 称为边 e 的**势差**，$\theta = (\theta_e)_{e\in E}$ 称为 G 的**势差**.

由定义，位势 π 是 V 到 R^n 的函数，势差 θ 是 E 到 R^m 的函数. 设 A 是图 G 的关联矩阵，则式(3.17)的矩阵形式是

$$\theta = -A^T\pi. \tag{3.18}$$

例 3.6 在例 3.4 的有向图 G 中，设位势为

$$\pi = (1,3,5,7,9)^T,$$

则由式(3.18)得 G 的势差

$$\theta = -A^T\pi = -\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 4 \\ 2 \\ 6 \\ 2 \\ -4 \\ 2 \end{pmatrix}.$$

定理 3.11 设 G 是 (n,m) 连通有向图，$\theta \in R^m$，则 θ 是势差的充要条件是，θ 正交于每

个圈向量 μ.

证　必要性　设 $C=[i_1,i_2,\cdots,i_k]$ 是一个圈，其圈向量是 μ，而 C 的顶点序列是 $j_1,j_2,\cdots,j_k$，则

$$\mu_{i_1}\theta_{i_1}=\pi_{j_2}-\pi_{j_1}$$
$$\mu_{i_2}\theta_{i_2}=\pi_{j_3}-\pi_{j_2}$$
$$\cdots\cdots$$
$$\mu_{i_k}\theta_{i_k}=\pi_{j_1}-\pi_{j_k}.$$

于是向量 μ 与 θ 的内积 $\langle\mu,\theta\rangle=0$.

充分性　用下列规则对所有顶点标号.

规则 1　任取顶点 i_0，标 i_0 为 $\pi_{i_0}=0$.

规则 2　设顶点 i 已标，j 未标，

(1) 若有 $e=(i,j)$，标 j 为 $\pi_j=\pi_i+\theta_e$;

(2) 若有 $e=(j,i)$，标 j 为 $\pi_j=\pi_i-\theta_e$.

由此，每个顶点都从 i_0 得到一个标号. 下面证明每个顶点的标号是唯一的.

设从 i_0 分别沿两条路 P_1 与 P_2 标到顶点 j，相应向量设为 μ^1 与 μ^2，则顶点 j 所得标号分别为 $\langle\mu^1,\theta\rangle$ 与 $\langle\mu^2,\theta\rangle$. 若 $\langle\mu^1,\theta\rangle\neq\langle\mu^2,\theta\rangle$，则

$$\langle\mu^1-\mu^2,\theta\rangle\neq 0. \tag{3.19}$$

因为 $\mu^1-\mu^2$ 是闭迹向量，而闭迹能分解为边不交的圈之并，所以从式(3.19)，必有圈向量 μ 使 $\langle\mu,\theta\rangle\neq 0$，矛盾.　证毕.

例 3.7　在例 3.4 的有向图 G 有圈 $C=[e_1,e_2,e_6,e_4]$，其圈向量是

$$\mu=(-1,1,0,-1,0,-1,0,0)^T,$$

另一方面，在例 3.6 中给出同一个有向圈 G 的势差

$$\theta=(2,6,4,2,6,2,-4,2)^T,$$

容易验证 $\langle\mu,\theta\rangle=0$.

定理 3.12　有向图的边割向量是势差.

证　设 C 是一个圈，$N(S)$ 是一个边割，则 C 与 $N(S)$ 的公共边成对出现，且有如下两种情形.

(1) 每对公共边在 $N(S)$ 中同向，在 C 中反向. 此时 $N(S)$ 的边割向量的相应分量为两个"+ 1"或两个"− 1"，C 的圈向量的相应分量为一个"+ 1"，一个"− 1".

(2) 每对公共边在 $N(S)$ 中反向，在 C 中同向. 此时 $N(S)$ 的边割向量的相应分量为一个"+ 1"，一个"− 1"，C 的圈向量的相应分量为两个"+ 1"或两个"− 1".

无论出现何种情形，边割向量与圈向量的内积都等于零，故边割向量是势差.　证毕.

定理 3.13　设 G 是 (n,m) 连通有向图，则 G 的每个势差能表为余圈基的线性组合.

证　设 θ 是 G 的一个势差，T 是 G 的生成树. 适当选取基本余圈的方向，使基本余圈向量 $\gamma^1,\cdots,\gamma^t$ 相应于树枝的分量构成单位矩阵. 考虑势差 θ'，

$$\theta'\triangleq\theta-\sum_{i\in E(T)}\theta_i\gamma^i, \tag{3.20}$$

则 $\theta'_i=0,i\in E(T)$. 设 θ' 相应的位势是 π'，即

$$\theta' = -A^T\pi'. \tag{3.21}$$

因为$\theta'|_T = 0$,所以对所有顶点$j \in V(G)$,π'_j是常数,故由式(3.21)得$\theta' = 0$. 再由式(3.20)有

$$\theta = \sum_{i \in E(T)} \theta_i \gamma^i.$$

证毕.

由定理3.13直接得出:

推论3.14 势差由生成树唯一决定.

§3.4 圈空间和余圈空间

设G是(n,m)连通有向图,并令

$$\Phi \triangleq \{\varphi \in R^m | \varphi \text{是} G \text{的环流}\},$$

$$\Theta \triangleq \{\theta \in R^m | \theta \text{是} G \text{的势差}\},$$

利用定理3.7容易得到:

(1) $(\varphi_1, \varphi_2) \in \Phi \implies (\varphi_1 + \varphi_2) \in \Phi$,

(2) $a \in R, \varphi \in \Phi \implies a\varphi \in \Phi$.

因此,Φ构成一个线性子空间. 同样,由定理3.11可得Θ也是一个线性子空间.

定理3.15 线性子空间Φ是由圈基生成的子空间. 线性子空间Θ是由余圈基生成的子空间.

证 由定理3.8,圈向量是环流. 又由定理3.9,每个环流能表为圈基的线性组合. 因此,Φ是由圈基生成的子空间. 同样由定理3.12和定理3.13,Θ是由余圈基生成的子空间. **证毕.**

由定理3.15可得如下定义:

定义3.8 设G是(n,m)连通有向图,则称Φ为G的**圈空间**,称Θ为G的**余圈空间**.

定理3.16 余圈空间Θ是关联矩阵A的行空间,即A的行向量生成的子空间.

证 由式(3.18),$\theta \in R^m$是势差的充要条件是,θ是关联矩阵A的行向量的线性组合. 故得证.

定理3.17 圈空间Φ和余圈空间Θ互为正交补.

证 设A是关联矩阵,则

$$\begin{aligned}
\varphi \in \Phi &\iff A\varphi = 0 && (\text{定理 } 3.7) \\
&\iff \langle \varphi, A_{i\cdot} \rangle = 0,\ \forall i \\
&\iff \langle \varphi, \theta \rangle = 0 && (\text{定理 } 3.16) \\
&\iff \theta \in \Theta. && (\text{定理 } 3.11)
\end{aligned}$$

故得证.

定理3.18 设G是(n,m)连通有向图,则线性子空间Φ的维数$\dim\Phi$与Θ的维数$\dim\Theta$满足

$$\dim\Phi + \dim\Theta = m. \tag{3.22}$$

证 由定理3.15,$\dim\Phi$即圈秩,$\dim\Theta$即余圈秩,再由定理3.1得式(3.22). **证毕.**

设 G 是 (n,m) 连通有向图，A 是关联矩阵，则由定理 3.7，

$$\Phi = \{\varphi \in R^m \mid A\varphi = 0\}.$$

因此，我们称 $\dim\Phi$ 为图 G 的零度. 又由式(3.18)，

$$\Theta = \{\theta = -A^T\pi \mid \pi \in R^n\}.$$

因此，我们称 $\dim\Theta$ 为图 G 的秩. 于是，从上面的讨论得到

图 G 的零度 = 圈空间的维数 = 圈秩；

图 G 的秩 = 余圈空间的维数 = 余圈秩 = 关联矩阵的秩.

定义 3.9 若一个矩阵的行向量是圈空间(余圈空间)的基，则称该矩阵为圈空间(余圈空间)的基矩阵.

由定理 3.15 和定理 3.16，基本圈矩阵是圈空间的基矩阵，基本余圈矩阵和基本关联矩阵都是余圈空间的基矩阵.

设 A 是一个矩阵，其列由图 G 的边集 E 标出. 若 $S\subseteq E$，则 $A|S$ 表示 A 的由 S 标出的子矩阵.

定理 3.19 设 $G=(V,E)$ 是 (n,m) 连通有向图，M 是 Θ 的基矩阵，C 是 Φ 的基矩阵，$S\subseteq E$，则

(1) $M|S$ 的列无关 $\Longleftrightarrow$ $G[S]$ 不含圈，

(2) $C|S$ 的列无关 $\Longleftrightarrow$ $G[S]$ 不含余圈.

证 (1) $M|S$ 的列相关，当且仅当存在 E 上函数 φ 使

$$\begin{cases} \varphi_e \neq 0,\ \exists\, e \in S, \\ \varphi_e = 0,\ \forall\, e \in E\backslash S, \\ \sum\limits_{e\in E} \varphi_e M_{\cdot e} = 0, \end{cases} \tag{3.23}$$

即存在 E 上函数 φ 使

$$\varphi \neq 0, \tag{3.24}$$

$$\varphi|_{E\backslash S} = 0, \tag{3.25}$$

$$M\varphi = 0. \tag{3.26}$$

由定理 3.17，式(3.26)表示 φ 是环流. 式(3.24)与(3.25)表示 $\varphi|S$ 是 $G[S]$ 的非零环流. 于是我们得到，$M|S$ 的列相关，当且仅当 $G[S]$ 有非零环流. 再由习题 3.4，$M|S$ 的列相关，当且仅当 $G[S]$ 含圈.

定理 3.19(2) 由读者自证(习题 3.15).

于是定理得证.

§3.5 应用——生成树的数目

定理 3.20 设 G 是 (n,m) 连通有向图，M 是 Θ 的基矩阵，C 是 Φ 的基矩阵，以 $\tau(G)$ 或 τ 表示 G 的生成树的数目，则

(1) $\tau(G) = M$ 的满阶非奇异子方阵的数目，

(2) $\tau(G) = C$ 的满阶非奇异子方阵的数目.

证 设 $S\subseteq E(G)$，则

(1) $M|S$ 是 M 的满阶非奇异子方阵

$\Longleftrightarrow G[S]$ 不含圈，$|S| = n-1$ （定理 3.19）

$\Longleftrightarrow G[S]$ 是生成树. （定理 2.1）

(2) $C|S$ 是 C 的满阶非奇异子方阵

$\Longleftrightarrow G[S]$ 不含余圈，$|S| = m-n+1$ （定理 3.19）

$\Longleftrightarrow G-S$ 连通，$|E(G-S)| = n-1$ （定理 1.11）

$\Longleftrightarrow G-S$ 生成树. （定理 2.1）

故定理得证.

设 G 是连通有向图，生成树 T 对应的基本圈矩阵是 C，基本余圈矩阵是 M. 我们总可以适当选择圈与余圈的方向，使基本圈矩阵相应于连枝的子矩阵是单位矩阵，基本余圈矩阵相应于树枝的子矩阵是单位矩阵. 在定理 3.9 和定理 3.13 的证明中就是这样作的. 凡连通有向图的基本圈矩阵和基本余圈矩阵，我们总作这样的假定，而不再加以说明.

定理 3.21　设生成树 T_i 的基本圈矩阵 C_i，基本余圈矩阵 M_i，$i=1,2$，则

(1) $M_2 = (M_2|T_1)M_1$，

(2) $C_2 = (C_2|\overline{T}_1)C_1$，

其中 $\overline{T}_1$ 是 T_1 的余树.

证　(1)　因为 M_1，M_2 都是余圈空间的基矩阵，所以有满秩阵 P 使 $M_2 = PM_1$. 于是

$$M_2|T_1 = (PM_1)|T_1 = P(M_1|T_1) = P,$$

$$M_2 = PM_1 = (M_2|T_1)M_1.$$

定理 3.21(2) 由读者自证(习题 3.18).

于是定理得证.

定义 3.10　称矩阵 A 是**单模的**，若 A 的满阶子式都是 0 或 1 或 -1. 称矩阵 A 是**全单模的**，若 A 的各阶子式都是 0 或 1 或 -1.

定理 3.22　有向图的基本圈矩阵和基本余圈矩阵都是全单模的.

证　基本圈矩阵的全单模性作为练习(习题 3.20). 今证基本余圈矩阵的全单模性. 为此，不妨设 G 是连通有向图，相应于生成树 T 的基本余圈矩阵是 M，P 是 M 的任一满阶子方阵. 由定理 3.20，有生成树 T_1 使 $P = M|T_1$. 设 M_1 是由生成树 T_1 决定的基本余圈矩阵，则由定理 3.21，

$$M = (M|T_1)M_1,$$

于是

$$M|T = (M|T_1)(M_1|T).$$

∵　$M|T$ 是单位阵

∴　$\det(M|T_1)\det(M_1|T) = 1$

∵　$M|T_1$ 与 $M_1|T$ 是整数矩阵

∴　$\det P = \det(M|T_1) = \pm 1$

∴　M 是单模的.

现在用归纳法证明 M 的全单模性. 设 P_k 是 M 的 k 阶子方阵. 当 $k=1$，因为 M 的元是 0 或 1 或 -1，所以 $\det P_k = 0$ 或 1 或 -1. 假设当 $k \leqslant t (t \geqslant 1)$ 时，$\det P_k = 0$ 或 1 或 -1.

今设 $k = t + 1$，有两种情形.

(1) P_{t+1} 含有 $M|T = I$(单位阵)的列，该列至多一个 1，其余的元都是 0. 若全为零，则 $\det P_{t+1} = 0$. 若恰有一个 1，则按该列展开，由归纳假设，$\det P_{t+1} = 0$ 或 1 或 -1.

(2) P_{t+1} 全由 $M|\overline{T}$ 的列组成，不妨设 P_{t+1} 在 M 的左上角，即

$$M \triangleq (M|\overline{T}, M|T) \triangleq \begin{pmatrix} P_{t+1} & * & \\ \circledast & *, & I \end{pmatrix},$$

令 $M|\overline{T}$ 的前 $(t+1)$ 列，加上 $M|T = I$ 的后 $(n-t-2)$ 列构成 M 的满阶子方阵 P，即

$$P = \begin{pmatrix} P_{t+1} & \\ \circledast, & I_{\cdot t+2}, \cdots, I_{\cdot n-1} \end{pmatrix},$$

由行列式的性质，有 $\det P = \det P_{t+1}$. 因为 M 是全单模的，所以 $\det P = 0$ 或 1 或 -1，故

$$\det P_{t+1} = 0 \text{ 或 } 1 \text{ 或 } -1.$$

证毕.

定理 3.23 设 G 是 (n,m) 连通有向图，相应于生成树 T 的基本圈矩阵是 B_f，基本余圈矩阵是 Q_f，则

$$\tau = \det B_f B_f^T = \det Q_f Q_f^T.$$

证 利用行列式的 Cauchy 公式：设 $A_{n\times m}$，$B_{m\times n}$，则

$$\det AB = \sum_i \det A_i \det B_i, \tag{3.27}$$

其中

A_i 是 A 的 n 列构成的 n 阶子方阵，

B_i 是 B 的相同标号的 n 行构成的 n 阶子方阵.

在式(3.27)中，用 Q_f 代 A，用 Q_f^T 代 B，则有

$$\det Q_f Q_f^T = \sum_{\substack{S\subseteq E \\ |S| = n-1}} (\det(Q_f|S))^2, \tag{3.28}$$

因为 Q_f 是单模的，所以

$$\det(Q_f|S) = \begin{cases} 0, & Q_f|S \text{ 奇异}, \\ \pm 1, & Q_f|S \text{ 非奇异}. \end{cases} \tag{3.29}$$

把式(3.29)代入式(3.28)即得

$$\det Q_f Q_f^T = \sum_{\substack{S\subseteq E \\ |S| = n-1 \\ Q_f|S\text{非奇异}}} (\det(Q_f|S))^2 = \tau. \quad (\text{由定理 } 3.20)$$

同样，由式(3.27)和 B_f 的单模性，可得

$$\det B_f B_f^T = \tau.$$

证毕.

定理 3.24 设 A_f 是 (n,m) 连通有向图的基本关联矩阵，则

$$\tau = \det A_f A_f^T.$$

证 因为 A_f 是余圈空间的基矩阵，由定理 3.20 有

$$\tau = A_f \text{ 的满阶非奇异于矩阵的数目}, \tag{3.30}$$

于是

$$\det A_f A_f^T = \sum_{\substack{S\subseteq E\\|S|=n-1}} (\det(A_f|S))^2 \qquad (\text{式}(3.27))$$

$$= \sum_{\substack{S\subseteq E\\|S|=n-1\\A_f|S\text{非奇异}}} \qquad (\text{习题 } 3.21)$$

$$= \tau. \qquad (\text{式}(3.30))$$

证毕.

§3.6 应用——矩阵之间的关系

设 G 是 (n,m) 连通图，其关联矩阵，基本关联矩阵，圈矩阵，基本圈矩阵，余圈矩阵和基本余圈矩阵分别是 A, A_f, B, B_f, Q 和 Q_f.

在 §3.1 指出：

$$B_f \overset{\text{对称差}}{\Longrightarrow} B,$$

$$Q_f \overset{\text{对称差}}{\Longrightarrow} Q,$$

相反的问题是利用对称差求 B_f 与 Q_f. 设 $S\subseteq E(G)$，则

$B_f|S$ 的列独立 $\Longleftrightarrow$ S 的边是连枝，（习题 3.33）

$Q_f|S$ 的列独立 $\Longleftrightarrow$ S 的边是树枝，（习题 3.31）

$A_f|S$ 的列独立 $\Longleftrightarrow$ S 的边是树枝.（习题 3.28）

因此，当

$$B \overset{\text{对称差}}{\Longrightarrow} \begin{pmatrix} I & * \\ 0 & 0 \end{pmatrix}$$

时，$(I \quad *)$ 即 B_f；当

$$A\text{ 或 }Q \overset{\text{对称差}}{\longrightarrow} \begin{pmatrix} I & * \\ 0 & 0 \end{pmatrix}$$

时，$(I \quad *)$ 即 Q_f.

由于 A_f 与 Q_f 都是余圈空间 Θ 的基矩阵，所以

$$A_f \overset{\text{对称差}}{\Longleftrightarrow} Q_f.$$

例 3.8 在例 3.2 中得到图 3.2(a) 的圈矩阵是

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix},$$

试求基本圈.

$$\because \quad B=\begin{array}{c}\begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8\end{matrix}\\ \begin{pmatrix}1&1&1&0&0&0&0&0\\0&0&0&1&1&0&1&0\\0&0&0&0&1&1&0&1\\0&0&0&1&0&1&1&1\end{pmatrix}\end{array}\oplus \Rightarrow \begin{pmatrix}1&1&1&0&0&0&0&0\\0&0&0&1&1&0&1&0\\0&0&0&0&1&1&0&1\\0&0&0&0&1&1&0&1\end{pmatrix}\oplus$$

$$\Rightarrow \begin{pmatrix}1&1&1&0&0&0&0&0\\0&0&0&1&1&0&1&0\\0&0&0&0&1&1&0&1\\0&0&0&0&0&0&0&0\end{pmatrix}.$$

$$\therefore \quad B_f=\begin{array}{c}\begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8\end{matrix}\\ \begin{pmatrix}1&1&1&0&0&0&0&0\\0&0&0&1&1&0&1&0\\0&0&0&0&1&1&0&1\end{pmatrix}\end{array}.$$

$\therefore$ 基本圈是 $C_1=[e_1,e_2,e_3]$，$C_2=[e_4,e_5,e_7]$，$C_3=[e_5,e_6,e_8]$.

定理 3.25 设 G 是 (n,m) 连通图，则在模 2 意义下

$$AB^T=0,$$

$$QB^T=0.$$

定理 3.25 的证明作为练习(习题 3.26 与 3.27).

定理 3.26 设 G 是 (n,m) 连通有向图，则

$$AB^T=0, \tag{3.31}$$

$$QB^T=0. \tag{3.32}$$

证 由定理 3.16，余圈空间 Θ 是 A 的行空间；由定理 3.15，Θ 是 Q 的行空间，圈空间 Φ 是 B 的行空间；由定理 3.17，Θ 与 Φ 互为正交补；于是式(3.31)与(3.32)成立． 证毕.

设 G 是 (n,m) 连通图，T 是生成树，$\overline{T}$ 是余树，则可把 A_f，B_f 与 Q_f 写为分块形式：

$$A_f \triangleq (A_f|\overline{T},A_f|T) \triangleq (A_{f,1},A_{f,2}), \tag{3.33}$$

$$B_f \triangleq (B_f|\overline{T},B_f|T) \triangleq (I_{m-n+1},B_{f,2}), \tag{3.34}$$

$$Q_f \triangleq (Q_f|\overline{T},Q_f|T) \triangleq (Q_{f,1},I_{n-1}). \tag{3.35}$$

定理 3.27 设 G 是 (n,m) 连通有向图.

(1) 当 G 是有向图，有

$$Q_{f,1}=-B_{f,2}^T=A_{f,2}^{-1}A_{f,1}, \tag{3.36}$$

$$Q_f=A_{f,2}^{-1}A_f. \tag{3.37}$$

(2) 当 G 是无向图，有

$$Q_{f,1}=B_{f,2}^T=A_{f,2}^TA_{f,1}, \tag{3.38}$$

$$Q_f=A_{f,2}^{-1}A_f. \tag{3.39}$$

证 对无向图的证明读者自证(习题 3.41). 今设 G 是有向图.

$$\because \quad 0=A_fB_f^T=(A_{f,1},A_{f,2})\begin{pmatrix}I_{m-n+1}\\B_{f,2}^T\end{pmatrix}=A_{f,1}+A_{f,2}B_{f,2}^T$$

$$\therefore \quad B_{f,2}^T=-A_{f,2}^{-1}A_{f,1}$$

$$\because \qquad 0 = Q_f B_f^T = (Q_{f,1}, I_{n-1})\begin{pmatrix} I_{m-n+1} \\ B_{f,2}^T \end{pmatrix} = Q_{f,1} + B_{f,2}^T$$

$$\therefore \qquad Q_{f,1} = -B_{f,2}^T = A_{f,2}^{-1} A_{f,1},$$

故式(3.36)得证. 将此结果代入式(3.35),有

$$Q_f = (A_{f,2}^{-1} A_{f,1}, I_{n-1}) = A_{f,2}^{-1}(A_{f,1}, A_{f,2}) = A_{f,2}^{-1} A_f,$$

故式(3.37)得证. 证毕.

习 题 三

3.1 在图 3.5 中给出图 G 的生成树 T,求基本圈矩阵和圈矩阵.

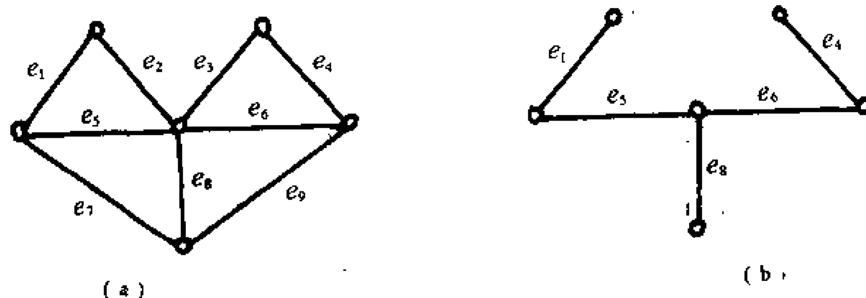


图 3.5 (a) 图 G (b) 生成树 T

3.2 在图 3.6 中给出图 G 和生成树 T,求基本余圈矩阵和余圈矩阵.

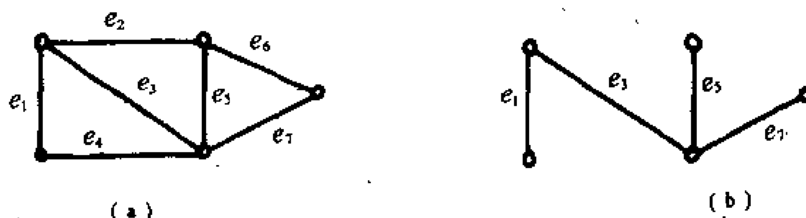


图 3.6 (a) 图 G (b) 生成树 T

3.3 设 G 连通有向图,若 G 没有有向余圈,则 G 有有向圈构成的圈基.

3.4 有向图 G 有圈的充要条件是 G 有非零环流.

3.5 设 G 连通有向图,则 G 的基础图是树的充要条件是 G 只有零环流.

3.6 在图 3.7 中给出环流 φ,生成树 T,试用基本圈向量表示环流 φ.

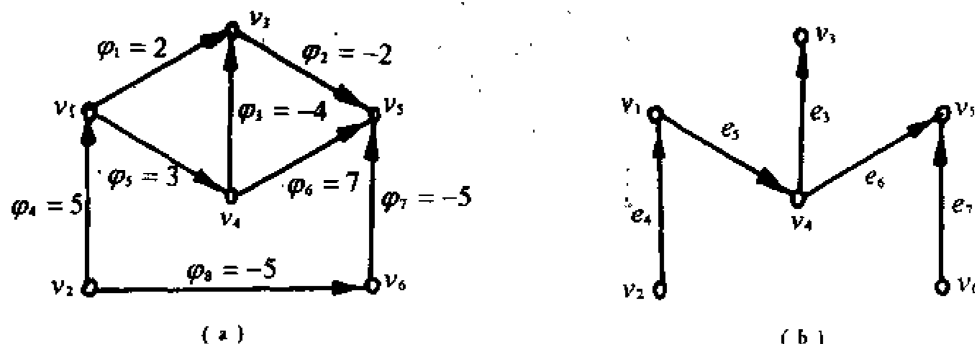


图 3.7 (a) 环流 φ (b) 生成树 T

3.7 设 G 是 (n,m) 有向图，则 $\varphi\in R^m$ 是非负环流的充要条件是，φ 能表为有向圈基的非负组合.

3.8 在图 3.8 中给出非负环流 φ，试把 φ 表为有向圈基的非负组合.

3.9 有向图 G 有余圈的充要条件是 G 有非零势差.

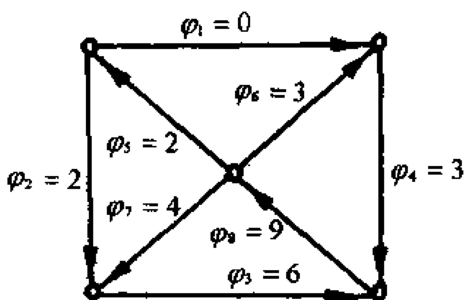
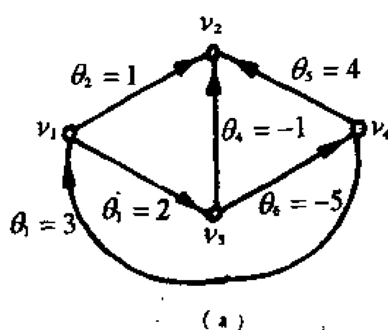


图 3.8

3.10 在图 3.9 中给出势差 θ，生成树 T，试用基本余圈向量表示势差 θ.



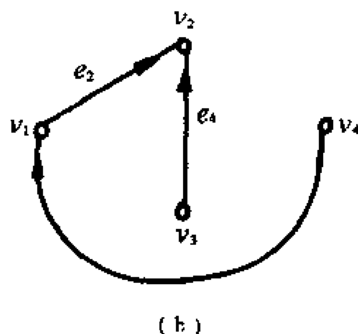


图 3.9 (a) 势差 θ (b) 生成树 T

3.11 设 G 是 (n,m) 有向图，则 $\theta\in R^m$ 是非负势差的充要条件是，θ 能表为有向余圈基的非负组合.

3.12 在图 3.10 中给出非负势差 θ，试把 θ 表为有向余圈基的非负组合.

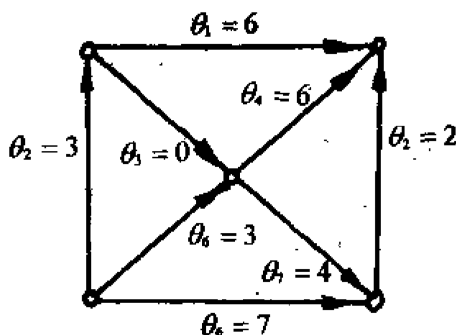


图 3.10

3.13　给定有向图 3.11 和余树 $\bar{T}$ 上的流量 $\varphi|_{\bar{T}}$，试扩充为图 G 的环流.

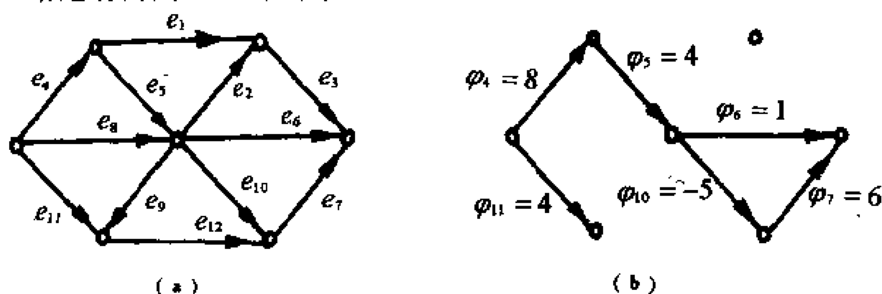


图 3.11　(a) 图 G　(b) 余树的流量 $\varphi|_{\bar{T}}$

3.14　给定有向图 3.12 和生成树 T 上的势差 $\theta|T$，试扩充为图 G 的势差.

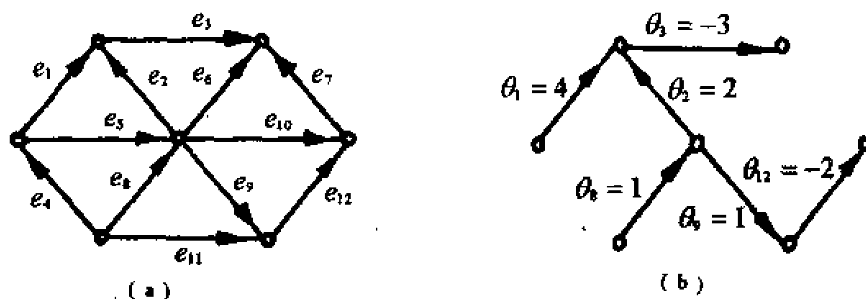


图 3.12　(a) 图 G　(b) 生成树的势差 $\theta|T$

3.15　证明定理 3.19 之(2).

3.16　如图 3.13，给定图 G 的生成树 T，利用定理 3.20(1) 求 G 的所有生成树.

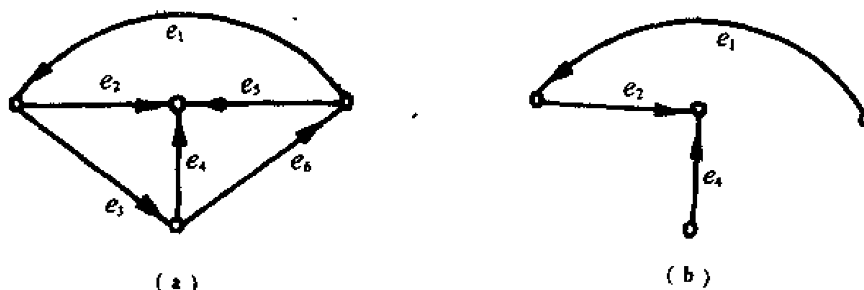


图 3.13　(a) 图 G　(b) 生成树 T

3.17　如习题 3.16，利用定理 3.20(2) 求 G 的所有生成树.

3.18　证明定理 3.21 之(2).

3.19　如习题 3.16，已知生成树 $T=[e_1,e_2,e_4]$，今再给定生成树 $T_1=[e_2,e_4,e_5]$，并以此验证定理 3.21 的结果.

3.20　试证基本圈矩阵是全单模的.

3.21　试证有向图的关联矩阵是全单模的.

3.22　由定理 3.23 和定理 3.24，分别利用基本圈矩阵，基本余圈矩阵和关联矩阵计算 $\tau(K_{3,3})$.

3.23　设 G 是简单图，试证 G 的关联矩阵 A 是全单模的，当且仅当 G 是偶图.

3.24　(1) 利用关联矩阵证明 $\tau(K_n)=n^{n-2}$，

(2) 画出 K_4 的所有生成树，验证(1).

3.25　(1) 利用关联矩阵证明 $\tau(K_{r,s})=r^{s-1}s^{r-1}$，

(2) 画出 $K_{2,4}$ 的所有生成树，验证(1).

3.26　设 (n,m) 连通图 G 的关联矩阵和圈矩阵分别是 A 和 B，则在模 2 意义下有

$$AB^T=0.$$

3.27　设 (n,m) 连通图 G 的圈矩阵和余圈矩阵分别是 B 和 Q，则在模 2 意义下有

$$QB^T=0.$$

3.28　设 (n,m) 连通图 G 的基本关联矩阵是 A_f，$S\subseteq E(G)$，则 $A_f|S$ 是满阶非奇异子方阵的充要条件是，$G[S]$ 是生成树.

3.29　设 (n,m) 连通图 G 的基本关联矩阵是 A_f，C 是闭迹，则 $A_f|C$ 的列相关.

3.30　利用习题 3.28 求 K_4 的所有生成树.

3.31　设 (n,m) 连通图 G 的基本余圈矩阵是 Q_f，$S\subseteq E(G)$，则 $Q_f|S$ 是满阶非奇异子方阵的充要条件是，$G[S]$ 是生成树.

3.32　利用习题 3.31 求 $K_{2,3}$ 的所有生成树.

3.33　设 (n,m) 连通图 G 的基本圈矩阵是 B_f，$S\subseteq E(G)$，则 $B_f|S$ 是满阶非奇异子方阵的充要条件是，$G[E\backslash S]$ 是生成树.

3.34　利用习题 3.33 求 $K_{2,3}$ 的所有生成树.

3.35　设连通有向图 G 的基本圈矩阵和基本余圈矩阵分别是 B_f 和 Q_f，则

$$\tau(G)=\pm\det\begin{pmatrix}B_f\\Q_f\end{pmatrix}.$$

3.36　设连通有向图 G 的基本关联矩阵和基本圈矩阵分别是 A_f 和 B_f，则

$$\tau(G)=\pm\det\begin{pmatrix}A_f\\B_f\end{pmatrix}.$$

3.37　试由下面的余圈矩阵 Q，

$$Q=\begin{array}{c}\\ F_1\\F_2\\F_3\\F_4\\F_5\\F_6\\F_7\\F_8\\F_9\\F_{10}\\F_{11}\\F_{12}\\F_{13}\end{array}\begin{array}{c}\begin{array}{cccccccc}e_1&e_2&e_3&e_4&e_5&e_6&e_7&e_8\end{array}\\ \begin{pmatrix}1&1&1&0&0&0&0&0\\1&0&0&1&1&0&0&0\\0&1&0&0&0&1&0&1\\0&0&0&0&1&0&1&1\\0&1&1&1&1&0&0&0\\1&0&1&0&0&1&0&1\\1&0&0&1&0&0&1&1\\0&1&0&0&1&1&1&0\\0&0&1&1&1&1&0&1\\0&1&1&1&0&0&1&1\\1&0&1&0&1&1&1&0\\1&1&0&1&0&1&1&0\\0&0&1&1&0&1&1&0\end{pmatrix}\end{array},$$

求基本余圈矩阵,并写出对应的生成树和基本余圈.

3.38　给出图 G 的关联矩阵 A 求基本余圈矩阵,并写出对应的生成树和基本余圈,这里

$$A=\begin{pmatrix}1&1&1&0&0&0&0&0\\1&0&0&1&1&0&0&0\\0&0&1&1&0&1&1&0\\0&1&0&0&0&1&0&1\\0&0&0&0&1&0&1&1\end{pmatrix}.$$

3.39　已知图 G 的圈矩阵 B,

$$B=\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \\ C_1 & 0&0&1&1&0&1&0&0&0\\ C_2 & 0&0&0&0&1&0&1&1&0\\ C_3 & 1&1&0&0&0&0&1&1&0\\ C_4 & 1&1&0&0&1&0&0&0&0\\ C_5 & 0&0&0&0&0&1&0&1&1\\ C_6 & 0&0&1&1&0&0&0&1&1\\ C_7 & 0&0&0&0&1&1&1&0&1\\ C_8 & 1&1&0&0&0&1&1&0&1\\ C_9 & 0&0&1&1&1&0&1&0&1\\ C_{10} & 1&1&1&1&0&0&1&0&1\end{matrix},$$

求基本圈矩阵,并写出对应的生成树和基本圈.

3.40　已知连通图 G 的基本余圈:

$$F_1=[e_1,e_4,e_7,e_8],$$
$$F_2=[e_2,e_6,e_8],$$
$$F_3=[e_3,e_4,e_6,e_7],$$
$$F_4=[e_5,e_7,e_8].$$

试求基本关联矩阵和关联矩阵.

3.41　证明定理 3.27 之(2).

3.42　由图 3.5 的无向图 G 和生成树 T,验证定理 3.27(2).

3.43　由图 3.7 的有向图 G 和生成树 T,验证定理 3.27(1).

第四章　匹配及其应用

§4.1　最大匹配

定义 4.1　设 $G=(V,E)$，$M\subseteq E$，若 M 的边互不相邻，则称 M 是 G 的一个**匹配**. M 的边称为**匹配边**，$E\backslash M$ 的边称为**自由边**. 若 $(u,v)\in M$，则称 u(或 v) 是 v(或 u) 的**配偶**. 若顶点 v 与 M 的一条边关联，则称 v 是 **M-饱和的**；否则称为 **M-非饱和的**. 设 M 是一个匹配，若每个顶点都是 M-饱和的，则称 M 是**完美匹配**. 设 M 是一个匹配，若不存在匹配 M' 使 $|M'|>|M|$，则称 M 为**最大匹配**.

由定义，在讨论图 G 的匹配时，只需考虑图 G 的基图.

例 4.1　从定义即得完美匹配必是最大匹配，而反之不真. 如图 4.1(a) 的匹配(匹配边用粗线表示，下同) 是最大匹配但不是完美匹配，实际上该图没有完美匹配. 图 4.1(b) 的匹配是完美匹配，自然也是最大匹配.

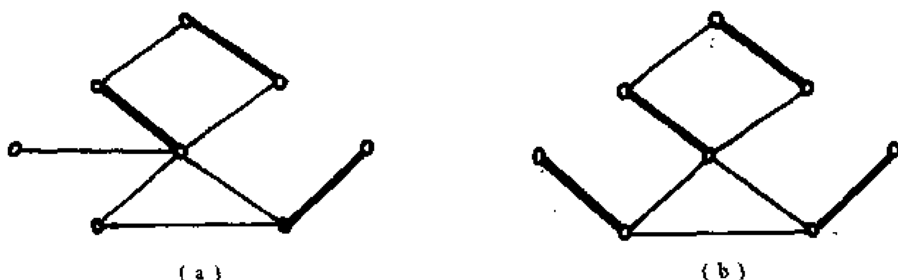


图 4.1　(a) 最大匹配　(b) 完美匹配

例 4.2　一飞行大队有 n 个飞行员，每架飞机需 2 名飞行员，由于技术水平等原因，有的飞行员不能同时驾一架飞机. 问在一次飞行中同时起飞的飞机最多多少架？

解　把飞行员作为顶点，两个顶点相邻，当且仅当两个飞行员能同时驾一架飞机. 由此得到图 G，问题转化为求 G 的最大匹配.

例 4.3　有 n 个人欲应聘 n 件工作. 每个人按自己的意愿至少选择一件工作应聘，问能否使每个人都按自己的意愿应聘？

解　把 n 个人作为顶点集 $X=\{x_1,\cdots,x_n\}$，把 n 件工作作为顶点集 $Y=\{y_1,\cdots,y_n\}$，x_i 与 y_j 相邻，当且仅当人 x_i 应聘了工作 y_j. 由此得到偶图 $G=(X,Y,E)$，问题转化为求 G 的完美匹配.

定理 4.1　设 M 与 M' 是图 G 的两个不同的匹配，令 $H\triangleq(V,M\oplus M')$，则 H 的连通支是下列三种情形之一.

(1) 是一个孤立点；

(2) 是一个圈，其边交错于 M 和 M'；

(3) 是一条路，其边交错于 M 和 M'.

证　每个 $v \in V$，v 至多与 M 或 M' 的一条边关联，故 $d_H(v) = 0$ 或 1 或 2.

若 $d_H(v) = 0$，则 v 是孤立点，即情形(1).

若 $d_H(v) = 1$ 或 2，则 H 的连通支是圈或路. 因为 M 的边互不相邻，M' 的边互不相邻，所以 H 中相邻的边必交错于 M 和 M'. 这就是情形(2) 或(3).　**证毕.**

定义 4.2　设 M 是图 $G = (V,E)$ 的匹配，称其边交错于 M 和 $E\backslash M$ 的路(圈) 为 M-交错路(圈). 起点和终点都是 M- 非饱和点的交错路称为 M- **增广路**.

定理 4.2　设 M 是图 G 的匹配，则 M 是最大匹配的充要条件是，G 没有 M- 增广路.

证　**必要性**　设 M 是最大匹配. 若存在 M- 增广路 P，则

$$M' \triangleq M \oplus P$$

是一个匹配且 $|M'| = |M| + 1$，与 M 是最大匹配相矛盾.

充分性　设 M 是匹配，G 不含 M- 增广路. 若 M 不是最大匹配，不妨设 M' 是最大匹配，则 $|M'| > |M|$. 由必要性，G 没有 M'- 增广路. 由定理 4.1，并注意到 $|M'| > |M|$，必有路 P，

$$P = v_0 v_1 \cdots v_{2m+1},$$

$$(v_0, v_1), (v_2, v_3), \cdots, (v_{2m}, v_{2m+1}) \in M',$$

$$(v_1, v_2), (v_3, v_4), \cdots, (v_{2m-1}, v_{2m}) \in M,$$

于是 P 是 M- 增广路，矛盾.　**证毕.**

定理 4.2 提供了求最大匹配的基本思想和方法. 若没有增广路，当前匹配就是最大匹配. 若有增广路，则可由对称差求得边数多一条的匹配. 继续这一过程，有限步之后必可求得最大匹配.

匹配与覆盖有密切的关系.

定义 4.3　设 $G = (V,E)$，$K \subseteq V$，若每条边都与 K 的一个顶点关联，则称 K 是图 G 的一个**覆盖**. 设 K 是一个覆盖，若不存在覆盖 K' 使 $|K'| < |K|$，则称 K 是一个**最小覆盖**.

定理 4.3　设 M 是图 G 的匹配，K 是覆盖，则

(1) $|M| \leqslant |K|$;　(4.1)

(2) 若 $|M| = |K|$，则 M 是最大匹配，K 是最小覆盖.

证　因为 M 的边都不相邻，所以 K 中至少要 $|M|$ 个点才能覆盖 M 中的所有边，故式(4.1) 成立. 设 M^* 是最大匹配，K^* 是最小覆盖，则由式(4.1) 有

$$|M| \leqslant |M^*| \leqslant |K^*| \leqslant |K|.$$

因为 $|M| = |K|$，故 $|M| = |M^*| = |K^*| = |K|$.　**证毕.**

§ 4.2　完美匹配

定义 4.4　设 G_1 是 G 的连通支，若 $|V(G_1)|$ 是奇(偶) 数，则称 G_1 为**奇(偶) 连通支**. 以 $\omega_0(G)$ 表 G 的奇连通支数.

定理 4.4　图 $G = (V,E)$ 有完美匹配的充要条件是

$$\omega_0(G - S) \leqslant |S|, \ S \subseteq V. \tag{4.2}$$

证　不妨设 G 是简单图.

必要性 设 M 是完美匹配，S 是 V 的真子图，G_i 是 $G-S$ 的奇连通支，$i=1,2,\cdots,r$. 因为 $|V(G_i)|$ 是奇数，M 是完美匹配，故必有 G_i 的顶点 u_i 与 S 的顶点 v_i 匹配，$i=1,2,\cdots,r$. 于是 $\omega_0(G-S)=r\leqslant|S|$.

充分性 设式(4.2)成立，但 G 没有完美匹配. 在式(4.2)中令 $S=\varnothing$，则 $\omega_0(G)=0$，故 $|V|=n$ 是偶数. 于是以 V 为顶点集的完全图有完美匹配. 设 G^* 是以 V 为顶点集的边数最多的没有完美匹配的简单图，则 G 是 G^* 的生成子图，且 G^* 不是完全图. 令

$$U\triangleq\{v\in V|d_{G^*}(v)=n-1\},$$

则 $U\subset V$. 下面证明 G^*-U 的连通支都是完全图. 不然，设 G^*-U 的连通支 G' 不是完全图，则有 $u,y\in V(G')$，$(u,y)\notin E(G')$. 设

$$P=uu_1\cdots u_ty,\quad(t\geqslant1)$$

是最短 u-y 路，因 P 是最短路，故 $(u,u_2)\notin E(G')$. 令 $v\triangleq u_1$，$w\triangleq u_2$（当 $t=1$ 时 $u_2=y$）. 即 $\exists\ u,v,w\in V(G')$，(u,v)，$(v,w)\in E(G')$，$(u,w)\notin E(G')$. 因为 $v\notin U$，且 G^* 不是完全图，所以存在 $x\in V(G^*-U)$，$(x,v)\notin E(G^*)$. 由 G^* 的定义推出：

(1) $G^*+(u,w)$ 有完美匹配 M_1，且 M_1 含 (u,w)，

(2) $G^*+(x,v)$ 有完美匹配 M_2，且 M_2 含 (x,v).

由定理 4.1，$H=(V,M_1\oplus M_2)$ 的连通支是孤立点或交错偶圈. 有两种情形.

情形 1 (x,v) 与 (u,w) 在 H 的不同连通支中. 此时，设 (x,v) 在 H 的连通支——交错偶圈 C 中，则

$$\{M_1\text{ 在 }C\text{ 的边}\}\cup\{M_2\text{ 不在 }C\text{ 的边}\}$$

构成 G^* 的完美匹配，且不含边 (x,v) 与 (u,w)，矛盾.

情形 2 (x,v) 与 (u,w) 在 H 的同一个连通支——交错偶圈 C 中. 此时，不妨设顶点 u，w，x 与 v 依次出现在 C，则

$$\{M_1\text{ 在 }C\text{ 上 }w\cdots xv\text{ 段的边}\}\cup\{(v,w)\}\cup\{M_2\text{ 不在 }C\text{ 上 }w\cdots xv\text{ 段的边}\}$$

构成 G^* 的完美匹配，且不含边 (x,v) 与 (u,w)，矛盾.

于是已证得 G^*-U 的连通支都是完全图.

$\because$ $G-U$ 是 G^*-U 的生成子图

$\therefore$ $\omega_0(G^*-U)\leqslant\omega_0(G-U)\leqslant|U|$

现在我们能构造 G^* 的一个完美匹配如下：

(1) G^*-U 的每个奇连通支的一个顶点与 U 的一个顶点匹配，余下的顶点自相匹配；

(2) G^*-U 的每个偶连通支的顶点自相匹配；

(3) 因为 $|V|$ 是偶数，所以 U 中剩下的顶点有偶数个，可以自相匹配.

这与 G^* 没有完美匹配矛盾. 证毕.

§4.3 偶图的匹配

前两节关于匹配的一般性质对偶图自然也是成立的. 但偶图的匹配还有自身的重要性质，这就是本节所要讨论的.

定理 4.5 设 $G=(X,Y,E)$ 是偶图，M 是匹配，K 是覆盖，则 M，K 分别是最大匹配、最小覆盖的充要条件是

$$|M| = |K|. \tag{4.3}$$

证　充分性　这是定理 4.3(2) 的特例，故成立.

必要性　设

$$U \triangleq \{u \in X \mid u \text{ 是 } M\text{- 非饱和点}\},$$

$$Z \triangleq \{z \in V \mid \text{存在 } z \text{ 到 } U \text{ 的顶点的交错路}\},$$

$$S \triangleq Z \cap X, T \triangleq Z \cap Y,$$

则

(1) T 的顶点都是 M- 饱和的(否则有 M- 增广路，矛盾)，

(2) $X \setminus S$ 的顶点都是饱和的(因为 $U \subseteq S$)，

(3) $T = \Gamma(S)$(因为 S 的顶点经 T 的顶点有交错路到 U 的顶点).

设 $K^* = (X \setminus S) \cup T$. 若 K^* 不是覆盖，则必有一条边其一个端点在 S 中而另一个端点在 $Y \setminus T$ 中，这与(3) 相矛盾. 所以 K^* 是覆盖. 由(1) 和(2)，K^* 的顶点都是 M- 饱和的，故 $|K^*| \leqslant |M|$. 再由式(4.1) 得 $|K^*| = |M|$，所以 K^* 是最小覆盖，故

$$|K| = |K^*| = |M|,$$

即式(4.3) 成立.　**证毕.**

定理 4.6　设 $G = (X, Y, E)$ 是偶图，则 G 有一个匹配饱和 X 的所有顶点的充要条件是

$$|\Gamma(S)| \geqslant |S|, \ S \subseteq X. \tag{4.4}$$

证　必要性　设匹配 M 饱和 X 的所有顶点，$S \subseteq X$. 因为 S 的顶点都是饱和的，所以 S 的顶点与 $\Gamma(S)$ 的不同顶点相匹配，故式(4.4) 成立.

充分性　设式(4.4) 成立而 G 没有匹配饱和 X 的所有顶点. 设 M 是最大匹配，则有 $u \in X$，u 是 M- 非饱和的. 令

$$Z = \{z \in V \mid \text{存在 } u\text{-}z \text{ 交错路}\},$$

$$S = Z \cap X, \ T = Z \cap Y,$$

则

(1) u 是 Z 的唯一非饱和点(否则有 M- 增广路，矛盾)，

(2) $|T| = |S| - 1$(由(1)，$S \setminus \{u\}$ 与 T 的顶点两两匹配)，

(3) $\Gamma(S) = T$(因为 S 的顶点经 T 的顶点有交错路连接 u).

由(2) 与(3) 有

$$|\Gamma(S)| = |T| = |S| - 1 < |S|,$$

这与式(4.4) 矛盾.　**证毕.**

定理 4.7　偶图 $G = (X, Y, E)$ 有完美匹配的充要条件是

$$|\Gamma(S)| \geqslant |S|, \ S \subseteq V. \tag{4.5}$$

证　必要性　设 M 是 G 的完美匹配，则对任意 $S \subseteq V$，有

$$|\Gamma_G(S)| \geqslant |\Gamma_M(S)| = |S|,$$

即式(4.5) 成立.

充分性　取 $S = X$，由式(4.5)，

$$|X| = |S| \leqslant |\Gamma(S)| \leqslant |Y|.$$

同理 $|Y| \leqslant |X|$，故 $|X| = |Y|$. 由定理 4.6，G 有匹配 M 饱和 X 的所有顶点，因为 $|X| = |Y|$，所以 M 是一个完美匹配. 证毕.

§4.4 应用——人员分配问题之一

例 4.3 的另一提法就是人员分配问题. 假设把 n 个人分配去作 n 件工作，问每个人都能分配去作自己能胜任的工作吗？正如在例 4.3 中指出的，这是求偶图 $G = (X,Y,E)$ 的完美匹配的问题. 因为 $|X| = |Y|$，所以求 G 的完美匹配就是求 G 的一个匹配饱和 X 的所有顶点. 由定理 4.6，这等价于式(4.4)成立. 由此便提供了解决人员分配问题的基本思想和方法.

设 $G = (X,Y,E)$ 是偶图，$|X| = |Y|$，M 是 G 的一个匹配. 若 X 的顶点都是 M-饱和的，则 M 是完美匹配. 否则，选 M-非饱和点 $u \in X$，有两种情形.

情形 1 找到一条从 u 出发的 M-增广路 P，令

$$\hat{M} = M \oplus P,$$

则 $|\hat{M}| = |M| + 1$，以 $\hat{M}$ 代替 M 重复进行.

情形 2 没有从 u 出发的 M-增广路，令

$$Z = \{z \in V \mid \text{存在 } u\text{-}z \text{ 交错路}\},$$

$$S = Z \cap X,\ T = Z \cap Y.$$

如定理 4.6 的充分性证明一样，可得

$$\Gamma(S) = T,\ |\Gamma(S)| = |S| - 1 < |S|,$$

因此，G 没有完美匹配.

上述过程实际上是一棵树生长过程.

定义 4.5 设 M 是偶图 $G = (X,Y,E)$ 的匹配，$u \in X$ 是 M-非饱和点. 称 $H \subseteq G$ 是以 u 为根的 **M-交错树**，如果满足

(1) $u \in V(H)$，

(2) $\forall\ v \in V(H)$，在 H 有唯一的 u-v 路是 M-交错路.

以上分析得到求偶图的完美匹配的算法.

算法 4.1 Hungarian 方法——求偶图的完美匹配

设 $G = (X,Y,E)$ 是偶图，$|X| = |Y|$，M 是一个匹配.

第 1 步 若 X 的顶点都是 M-饱和的，则 M 是完美匹配. 否则取 M-非饱和点 $u \in X$，令

$$S = \{u\},\ T = \varnothing.$$

第 2 步 若 $\Gamma(S) = T$，则

$$|\Gamma(S)| = |T| = |S| - 1 < |S|,$$

因此 G 没有完美匹配. 否则取 $y \in \Gamma(S) \setminus T$. 若 y 是 M-饱和点，转第 3 步，否则转第 4 步.

第 3 步 设 $(y,x) \in M$，则令

$$S = S \cup \{x\},\ T = T \cup \{y\},$$

转第 2 步.

第 4 步　u-y 路是 M- 增广路，记为 P，并令

$$M = M \oplus P.$$

转第 1 步.

例 4.4　如图 4.2(a) 的偶图 G 和匹配 M，求完美匹配.

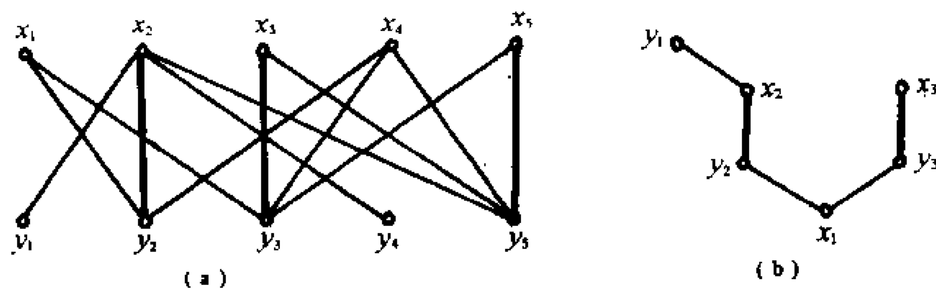


图 4.2　(a) 图 G 和匹配 M　(b) 以 x_1 为根的交错树

第 1 次迭代(如图 4.2)

1. $S = \{x_1\}, T = \varnothing$
2. $y_2 \in \Gamma(S) \backslash T$
3. $(y_2, x_2) \in M$，令 $S = \{x_1, x_2\}, T = \{y_2\}$
2. $y_3 \in \Gamma(S) \backslash T$
3. $(y_3, x_3) \in M$，令 $S = \{x_1, x_2, x_3\}, T = \{y_2, y_3\}$
2. $y_1 \in \Gamma(S) \backslash T$
4. $P = x_1 y_2 x_2 y_1$，

令

$$M = M \oplus P = \{(x_1, y_2), (x_2, y_1), (x_3, y_3), (x_5, y_5)\}$$，如图 4.3(a).

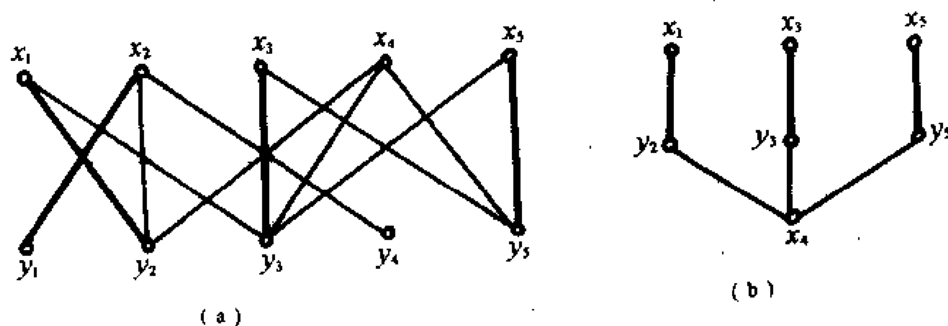


图 4.3　(a) 图 G 和匹配 M　(b) 以 x_4 为根的交错树

第 2 次迭代(如图 4.3)

1. $S = \{x_4\}, T = \varnothing$
2. $y_2 \in \Gamma(S) \backslash T$
3. $(y_2, x_1) \in M$，令 $S = \{x_4, x_1\}, T = \{y_2\}$

2. $y_3 \in \Gamma(S)\backslash T$

3. $(y_3,x_3) \in M$，令 $S = \{x_4,x_1,x_3\}$，$T = \{y_2,y_3\}$

2. $y_5 \in \Gamma(S)\backslash T$

3. $(y_5,x_5) \in M$，令 $S = \{x_4,x_1,x_3,x_5\}$，$T = \{y_2,y_3,y_5\}$

2. $\Gamma(S) = T$，

所以没有完美匹配.

注　在例 4.4 中选取 $y \in \Gamma(S)\backslash T$ 的方法是：依进入 S 的顶点的顺序，例如 $S = \{x_{i_1},\cdots,x_{i_t}\}$，则依次考查 $\Gamma(x_{i_1})\backslash T,\Gamma(x_{i_2})\backslash T,\cdots,\Gamma(x_{i_t})\backslash T$. 在第 1 次迭代中，$y_2 \in \Gamma(x_1)\backslash T$，$y_3 \in \Gamma(x_1)\backslash T$，至此对顶点 x_1 已考查完毕，转而考查顶点 x_2，得 $y_1 \in \Gamma(x_2)\backslash T$.

§4.5　应用——人员分配问题之二

人员分配问题之一是人员数与工作数相等，在这种情形，或者求出完美匹配，或者没有完美匹配. 在一般情形，人员数与工作数是不相等的，这就是人员分配问题之二.

例 4.5　设有 m 个人分配作 n 件工作，问如何分配使这 m 个人尽可能多地作他们胜任的工作. 我们把人员作为顶点集 X，把工作作为顶点集 Y. $x \in X$ 与 $y \in Y$ 相邻，当且仅当人员 i 胜任工作 j. 这就得到一个偶图，问题是求这个偶图的最大匹配.

一个图的完美匹配不一定存在，但一个图的最大匹配一定存在.

在算法 4.1 中任选 X 的非饱和点 u，当没有从 u 出发的增广路时，即可判定没有完美匹配而终止. 但在求最大匹配时，由定理 4.2，应考查 X 的所有非饱和点. 即查考 $X\backslash S$ 是否空集，若 $X\backslash S$ 非空，是否有非饱和点. 因此修改求偶图的完美匹配的算法 4.1，即可得求最大匹配的算法.

算法 4.2　求偶图的最大匹配的算法

设 $G = (X,Y,E)$ 是偶图，M 是一个匹配.

第 1 步　令 $S = \varnothing$，$T = \varnothing$.

第 2 步　若 M 饱和 $X\backslash S$ 的每个顶点，则 M 是最大匹配. 否则取非饱和点 $u \in X\backslash S$，令 $S = S \cup \{u\}$.

第 3 步　若 $\Gamma(S) = T$，转第 6 步. 否则取 $y \in \Gamma(S)\backslash T$. 若 y 是 M 饱和点，转第 4 步，否则转第 5 步.

第 4 步　设 $(y,x) \in M$，则令

$$S = S \cup \{x\},\ T = T \cup \{y\},$$

转第 3 步.

第 5 步　u-y 路是 M- 增广路，设为 P，并令

$$M = M \oplus P,$$

转第 1 步.

第 6 步　若 $X\backslash S = \varnothing$，则 M 是最大匹配，否则转第 2 步.

§ 4.6　应用——最优分配问题

在人员分配问题中若再考虑工作效率，问应如何分配工作使总的工作效率最大，这就是最优分配问题.

设 $X=\{x_1,\cdots,x_n\}$ 表人员的集合，$Y=\{y_1,\cdots,y_n\}$ 表工作的集合，w_{ij} 表人员 i 作工作 j 的效率. 相应作出赋权完全偶图 $G=(X,Y,E)$，w_{ij} 作为边 (x_i,y_j) 的权，则最优分配问题转化为求 G 的权最大的完美匹配.

定义 4.6　设 $G=(X,Y,E)$ 是赋权完全偶图，边 $e=(x,y)$ 的权记为 $w(e)$ 或 $w(x,y)$. 设 M 是 G 的匹配，则

$$w(M)=\sum\{w(e)\,|\,e\in M\} \tag{4.6}$$

称为 M 的**权**. 权最大的完美匹配称为**最优匹配**.

定义 4.7　设赋权完全偶图 $G=(X,Y,E)$ 的每个顶点 $v\in V$ 对应一个实数 $l(v)$，满足

$$l(x)+l(y)\geqslant w(x,y),\ x\in X,y\in Y, \tag{4.7}$$

则称 l 为 G 的一个**可行顶点标号**. 设 l 为可行顶点标号，则称生成子图 $G_l=(V,E_l)$ 为相应于可行顶点标号的**相等子图**，这里

$$E_l=\{(x,y)\in E\,|\,l(x)+l(y)=w(x,y)\}. \tag{4.8}$$

定理 4.8　设 l 是赋权完全偶图 $G=(X,Y,E)$ 的可行顶点标号，G_l 是相等子图，若 M^* 是 G_l 的完美匹配，则 M^* 是 G 的最优匹配.

证　因为 G_l 是 G 的生成子图，所以 M^* 也是 G 的完美匹配. 因为 G_l 是相等子图，所以由式(4.6)与式(4.8)有

$$w(M^*)=\sum\{l(v)\,|\,v\in V\}.$$

另一方面，对 G 的任一完美匹配 M，由式(4.7)有

$$w(M)\leqslant\sum\{l(v)\,|\,v\in V\},$$

故 M^* 是最优匹配.　**证毕.**

由定理 4.8 得到求最优匹配的基本思想和方法. 假定 l 是可行顶点标号，在相等子图 G_l 中用算法 4.1 求完美匹配. 若找到 G_l 的完美匹配，此即是所求最优匹配. 若 G_l 没有完美匹配，则可修改可行顶点标号，重复上述过程，直到找出新的相等子图的完美匹配.

初始可行顶点标号是容易得到的. 例如设

$$\begin{cases} l(x)=\max\ \{w(x,y)\,|\,y\in Y\},x\in X, \\ l(y)=0 \qquad\qquad\qquad\qquad ,y\in Y. \end{cases} \tag{4.9}$$

算法 4.3　可行顶点标号法——求赋权完全偶图的最优匹配

设 $G=(X,Y,E)$，$|X|=|Y|$，每条边 $e=(x,y)$ 有权 $w(e)$（也记为 $w(x,y)$），l 是一个可行顶点标号，M 是相等子图 G_l 的一个匹配.

第 1 步　若 X 的顶点都是饱和的，则 M 是最优匹配. 否则取 M- 非饱和点 $u\in X$，令

$$S=\{u\},\ T=\varnothing.$$

第 2 步　若 $\Gamma_{G_l}(S)=T$，则 G_l 没有完美匹配，转第 3 步，否则转第 4 步.

第 3 步　调整标号，计算

$$\alpha_l = \min\{l(x) + l(y) - w(x,y) \mid x \in S, y \notin T\}, \tag{4.10}$$

由此得新的可行顶点标号

$$l'(v) = \begin{cases} l(v) - \alpha_l, v \in S, \\ l(v) + \alpha_l, v \in T, \\ l(v) \qquad ,\text{否则}, \end{cases} \tag{4.11}$$

令 $l = l', G_l = G_{l'}$.

第 4 步　取 $y \in \Gamma_{G_l}(S) \backslash T$. 若 y 是 M- 饱和点，转第 5 步，否则转第 6 步.

第 5 步　设 $(y,x) \in M$，则令

$$S = S \cup \{x\},\ T = T \cup \{y\},$$

转第 2 步.

第 6 步　在 G_l 中的 u-y 交错路是 M- 增广路，记为 P，并令

$$M = M \oplus P,$$

转第 1 步.

注　在第 4 步取 $y \in \Gamma_{G_l}(S) \backslash T$ 的合理性的证明，有两种情形：

(1) 从第 2 步到第 4 步，自然可选出 y；

(2) 从第 3 步到第 4 步，必须证明选出 y 的合理性，即需证 $T \subset \Gamma_{G_{l'}}(S)$. 为此分如下三步来证明.

第 1 步证明 $\alpha_l > 0$. 设 $x \in S, y \notin T = \Gamma_{G_l}(S)$，则 $(x,y) \notin E_l$，故 $l(x) + l(y) > w(x,y)$，从而 $\alpha_l > 0$；

第 2 步证明 $T \subseteq \Gamma_{G_{l'}}(S)$. 设 $y \in T = \Gamma_{G_l}(S)$，则有 $x \in S, (x,y) \in E_l$，故 $l(x) + l(y) = w(x,y)$. 另一方面由式(4.11)，

$$\begin{aligned} l'(x) + l'(y) &= (l(x) - \alpha_l) + (l(y) + \alpha_l) \\ &= l(x) + l(y) = w(x,y), \end{aligned}$$

故 $(x,y) \in E_{l'}$，因此 $y \in \Gamma_{G_{l'}}(S)$；

第 3 步证明 $T \neq \Gamma_{G_{l'}}(S)$. 设 $x \in S, y \notin T$ 满足式(4.10)，则由式(4.11)，

$$l'(x) + l'(y) = (l(x) - \alpha_l) + l(y) = w(x,y),$$

故 $(x,y) \in E_{l'}$，因此 $y \in \Gamma_{G_{l'}}(S)$.

至此已证得 $T \subset \Gamma_{G_{l'}}(S)$.　证毕.

本节最后我们指出，用矩阵 W 表示一个赋权完全偶图是方便的，即

$$W = (w_{ij}) = \begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & \cdots & y_n \end{array} \\ \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix} \end{array},$$

其中 $w_{ij} = w(x_i, y_j)$.

例 4.6　求下列由矩阵 W 表示的赋权完全偶图的最优匹配.

$$W = \begin{matrix} & y_1 & y_2 & y_3 & y_4 \\ x_1 & 4 & 5 & 5 & 1 \\ x_2 & 2 & 2 & 4 & 6 \\ x_3 & 4 & 2 & 3 & 3 \\ x_4 & 5 & 0 & 2 & 1 \end{matrix}.$$

由式(4.9)得初始可行顶点标号如表 4.1. 右上角打"＊"的元表示

$$l(x) + l(y) = w(x,y).$$

作出相等子图 G_l,并任给 G_l 的匹配 M,如图 4.4.

表 4.1

$l(x)$ \ $l(y)$	0	0	0	0
5	4	5*	5*	1
6	2	2	4	6*
4	4*	2	3	3
5	5*	0	2	1

表 4.2

$l(x)$ \ $l(y)$	1	0	0	0
5	4	5*	5*	1
6	2	2	4	6*
3	4*	2	3*	3*
4	5*	0	2	1

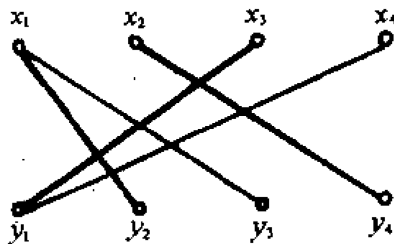


图 4.4 G_l 与匹配 M

1. $S = \{x_4\}, T = \varnothing$
4. $y_1 \in \Gamma_{G_l}(S) \backslash T$
5. $(y_1, x_3) \in M, S = \{x_4, x_3\}, T = \{y_1\}$
2. $\Gamma_{G_l}(S) = T$
3. $\alpha_l = 1$,按式(4.11)得新的可行顶点标号如表 4.2,相应作出新的相等子图而匹配 M 没有改变,如图 4.5. 继续迭代:

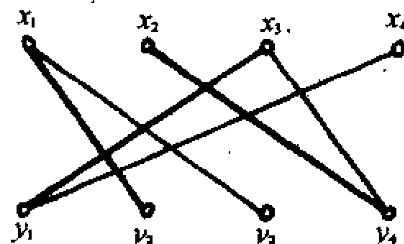


图 4.5 G_l 与匹配 M

4. $y_3 \in \Gamma_{G_l}(S) \backslash T$
6. $P = x_4 y_1 x_3 y_3$

$$M = M \oplus P = \{(x_1, y_2), (x_2, y_4), (x_3, y_3), (x_4, y_1)\},$$

如图 4.6.

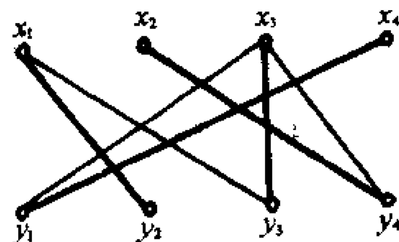


图 4.6 G_l 与匹配 M

1. X 的顶点都是饱和的,所以当前匹配为最优匹配,即

$$M = \{(x_1, y_2), (x_2, y_4), (x_3, y_3), (x_4, y_1)\},$$

其权为

$$w(M) = w_{12} + w_{24} + w_{33} + w_{41} = 19.$$

§ 4.7 应用——配对问题

例 4.2 的问题称为配对问题,即求图 G 的最大匹配的问题. 在 § 4.5 中所讨论的分配问题是求偶图的最大匹配,它显然是配对问题的特例. 从偶图推广到非偶图,问题变得更复杂更困难.

假定我们欲求图 G 的最大匹配. 对当前匹配 M 不妨假设:

(1) G 至少有两个非饱和点(否则,由习题 4.1,M 是最大匹配), (4.12)

(2) 非饱和点互不相邻(否则,在 M 中至少可增加一条匹配边). (4.13)

今作有向图 $\bar{G} = (V, \bar{E})$, $(x,y) \in \bar{E}$ 当且仅当存在 $u \in V$ 使 $(x,u) \in E \backslash M$, $(u,y) \in M$, 并称 $\bar{G}$ 为图 G 和匹配 M 的导出有向图. 容易看到在 G 中以 $E \backslash M$ 的边开始的偶交错路 $v_1v_2v_3 \cdots v_{2m+1}$, 对应在 $\bar{G}$ 中的有向路 $v_1v_3v_5 \cdots v_{2m+1}$. 反过来,在 $\bar{G}$ 中的有向路可能对应 G 中的偶交错通道或偶交错路.

定义 4.8 设 $\bar{G}$ 是图 G 和匹配 M 的导出有向图,称 $\bar{G}$ 中的有向路是合理的,如果它在 G 中对应一条偶交错路;否则称为不合理的.

例 4.7 在图 4.7(a) 中给出图 G 和满足式(4.12) 与式(4.13) 的匹配 M. 图 G 和匹配 M 的导出有向图 $\bar{G}$ 如图 4.7(b).

(a) (b)

图 4.7 (a) 图 G 和匹配 M (b) 导出有向图 $\bar{G}$

在 $\bar{G}$ 中的有向路 $v_7v_5v_2$ 是合理的,因为它对应 G 中的偶交错路 $v_7v_6v_5v_3v_2$.

在 $\bar{G}$ 中的有向路 $v_7v_6v_8v_5$ 是不合理的,因为它对应 G 中的偶交错通道 $v_7v_5v_6v_7v_8v_6v_5$.

定理 4.9 设 M 是 G 的一个匹配且满足式(4.12) 与式(4.13),$\bar{G}$ 是 G 和 M 的导出有向图,则 M 是最大匹配的充要条件是,在 $\bar{G}$ 中不存在从一个 M-非饱和点到另一个 M-非饱和点邻域的合理路.

证 在 G 的 M-增广路对应在 $\bar{G}$ 的从一个 M-非饱和点到另一个 M-非饱和点邻域的合理路,再由定理 4.2 得证.

定理 4.9 提供了求最大匹配的算法.

算法 4.4 合理路——求图的最大匹配的算法

设图 $G = (V, E)$, M 是满足式(4.13) 的匹配.

第 1 步 若不满足式(4.12),则 M 是最大匹配.

第 2 步 作 G 和 M 的导出有向图 $\bar{G}$.

第 3 步 在 $\bar{G}$ 不存在从一个 M-非饱和点到另一个 M-非饱和点邻域的合理路,则 M 是最大匹配.

第 4 步 设 $\bar{P}$ 是 $\bar{G}$ 中从一个 M-非饱和点到另一个 M-非饱和点邻域的合理路,$\bar{P}$ 对应于 G 中 M-增广路 P,则令

$$M = M \oplus P,$$

转第 1 步.

例 4.8　在图 4.7(a) 给出图 G 和匹配 M,求最大匹配.

首先作 G 和 M 的导出有向图,如图 4.7(b). 在 $\overline{G}$ 中任选非饱和点 v_1,搜索从 v_1 出发的所有有向路,得以 v_1 为根的外向树,如图 4.8(a).

在以 v_1 为根的外向树考查是否有 v_1 到另一个非饱和点 v_4 的邻域的合理路,即考查 v_1-v_2 路与 v_1-v_7 路. v_1-v_2 路 $v_1v_3v_5v_2$ 对应 G 中偶交错通道

$$v_1v_2v_3v_6v_5v_3v_2, \tag{4.14}$$

故是不合理的. v_1-v_7 路 $v_1v_3v_6v_7$ 对应 G 中偶交错路

$$v_1v_2v_3v_5v_6v_8v_7,$$

故是合理的,相应得到 G 的 M- 增广路 P,

$$P = v_1v_2v_3v_5v_6v_8v_7v_4.$$

令

$$M = M \oplus P = \{(v_1,v_2),(v_3,v_5),(v_6,v_8),(v_7,v_4)\},$$

如图 4.8(b). 因为式(4.12) 不满足,所以当前匹配为最大匹配.

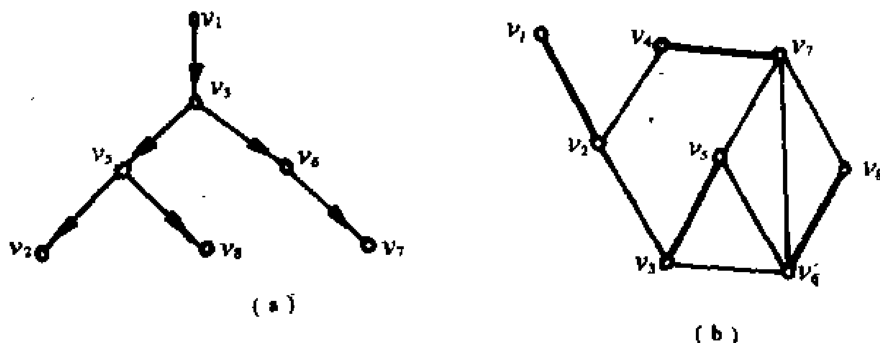


图 4.8　(a) 以 v_1 为根的外向树　(b) 最大匹配

算法 4.4 的关键步骤是判断 $\overline{G}$ 中有向路是否合理路. 注意到在图 4.8(a) 中有向路 $v_1v_3v_5v_2$ 是不合理的,因为它对应 G 中交错通道(4.14). 其本质的原因是在图 4.7(a) 从非饱和点 v_1 出发搜索是否有增广路的过程中,交错通道(4.14) 含了奇圈 $v_3v_6v_5v_3$. 为了处理好奇圈,引入下列定义.

定义 4.9　设 M 是图 $G=(V,E)$ 的匹配,若有圈 F, $|F|=2k+1(k\geqslant 1)$,且 F 含 k 条匹配边,则称 F 为一朵花,花中有唯一顶点不与花中其余顶点匹配,这个顶点称为花蒂.

定义 4.10　设图 $G=(V,E)$, $H\subseteq G$,则图

$$G/H = (V/H, E/H)$$

称为收缩图,其中

$$V/H = (V\backslash V(H)) \cup \{h\}.$$

而 E/H 的边有下列两类,且当有平行边时只保留一条边:

(1) 当 $i,j \in V\backslash V(H)$ 时

$$(i,j)\in E/H \Longleftrightarrow (i,j)\in E.$$

(2) 当 $i \in V\backslash V(H)$ 时

$$(i,h) \in E/H \Longleftrightarrow \exists \ j \in V(H), (i,j) \in E.$$

例 4.9 在图 4.7(a) 的图 G 和匹配 M，有花

$$F = v_3v_6v_5v_3,$$

今收缩花 F.

(1) 用一个顶点 f 代替花 F.

(2) 若 G 的边其端点都不在花 F 上，则仍作为收缩图的边，即有边 (v_1,v_2)，(v_2,v_4)，(v_4,v_7) 与 (v_7,v_8).

表 4.3

在 G 中	(v_2,v_3)	(v_5,v_7)	(v_6,v_7)	(v_6,v_8)
在 G/F 中	(v_2,f)	(f,v_7)	(f,v_7)	(f,v_8)

(3) 若 G 的边其端点一个不在花 F 上，另一个在花 F 上，则在收缩图中作为关联于 f 的边，其对应关系如表 4.3.

在 G/F 中去掉平行边 (f,v_7)，于是得收缩图 G/F，如图 4.9(a). 此图中有花 $F_1 = fv_8v_7f$，类似可得收缩图 $G/F/F_1$，如图 4.9(b).

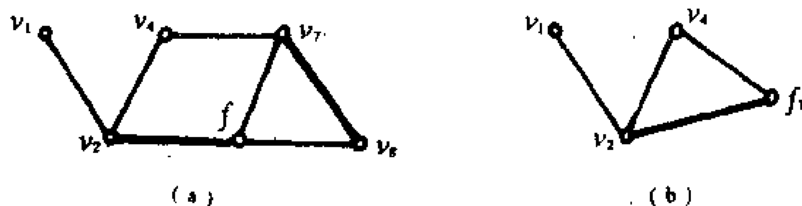


图 4.9 (a) 收缩图 G/F (b) 收缩图 $G/F/F_1$

在图 4.9(b) 中虽然也有花 $F_2 = v_2f_1v_4v_2$，但在搜索是否有增广路的过程中不会产生困难，因为从非饱和点 v_1 进入花 F_2 时并不经过花蒂 v_4. 换言之，困难的产生在于在搜索中当进入花时经过了花蒂，如图 4.7(a) 与图 4.9(a). 在这种情形就必须收缩花. 对花的收缩有两个问题.

问题 1 如何发现花.

问题 2 收缩花后增广路会不会增加或减少.

关于花的发现，考查图 4.10(a) 的花 F，在导出有向图中的搜索如图 4.10(b). 当从顶点 u_{2k-2} 搜索时将得到 u_{2k}，图中用虚线表示，这意味着匹配边 (u_{2k-1},u_{2k}) 使用了两次，从而发现了花.

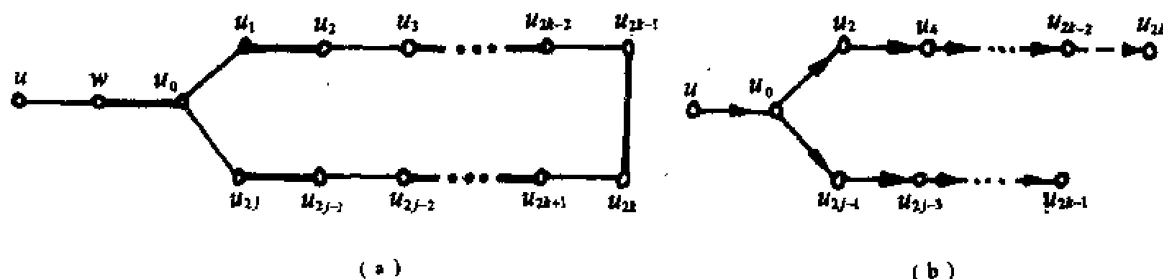


图 4.10 (a) 花 (b) 花的发现

关于问题 2，我们有下面两个定理.

定理 4.10 假设从 G 中非饱和点 u 出发搜索关于匹配 M 的增广路时发现了花 F，则对 F 中任一顶点 v，在 G 中都有 u-v 交错路，且其最后一条边是匹配边.

证 因为花 F 是从非饱和点 u 出发发现的，所以有 u 到 F 的花蒂的交错路. 故对花 F 中任一顶点 v，都有两条 u-v 交错路，其中必有一条交错路，它的最后一条边是匹配边. 证毕.

定理 4.11 假设从 G 中 M-非饱和点 u 出发搜索关于匹配 M 的增广路时发现了花 F，则在 G 中有从 u 出发的 M-增广路的充要条件是，在收缩图 G/F 中有从 u 出发的 M/F 增广路（也许 u 就是 F 的花蒂），其中

$$M/F \triangleq M \backslash M_F,\ M_F \triangleq M \cap F.$$

证 当增广路不过花时，G 与 G/F 中的增广路是一致的. 因此下面假定增广路过花.

必要性 设 P 是 G/F 中的 u-v_0 增广路且过顶点 f. 从 u 沿 P 到 f，f 的紧前顶点设为 x，f 的紧后顶点设为 y，P 的 u-x 段设为 P'，P 的 y-u_0 段设为 P''，则

$$P = P' + (x,f) + (f,y) + P''$$

有两种情形.

(1) (x,f) 是匹配边，则 (f,y) 不是匹配边. 如图 4.11(a) 是 G/F 中的增广路 P，(b) 是打开花 F 后在 G 中相应的部分. 容易看到，

$$\overline{P} \triangleq P' + (x,u') + P''' + (u'',y) + P''$$

是 G 中增广路，其中 (x,u') 对应 (x,f)，(u'',y) 对应 (f,y)，P''' 是在花上的 u'-u'' 段.

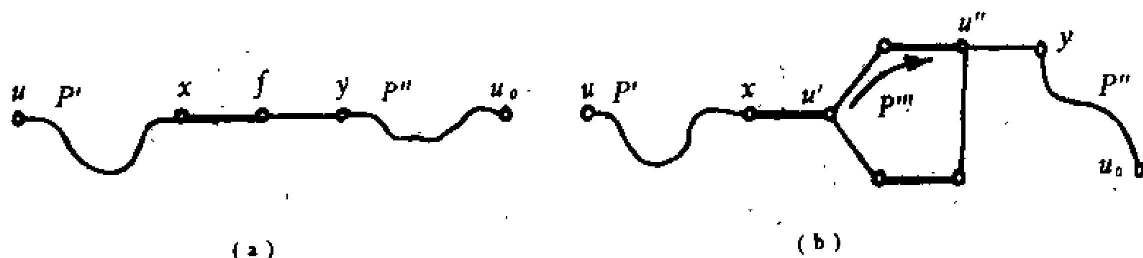


图 4.11 (a) G/F 中增广路 (b) G 中增广路

(2) (x,f) 不是匹配边，则 (f,y) 是匹配边，类似可从 G/F 中的增广路找出 G 中增广路.

充分性 设 P 是 G 中的 u-u_0 增广路且过花 F，则有三种情形.

(1) P 以匹配边进入 F，以自由边离开 F，如图 4.12(a). 此时收缩 F 即得 G/F 的增广路，如(b).

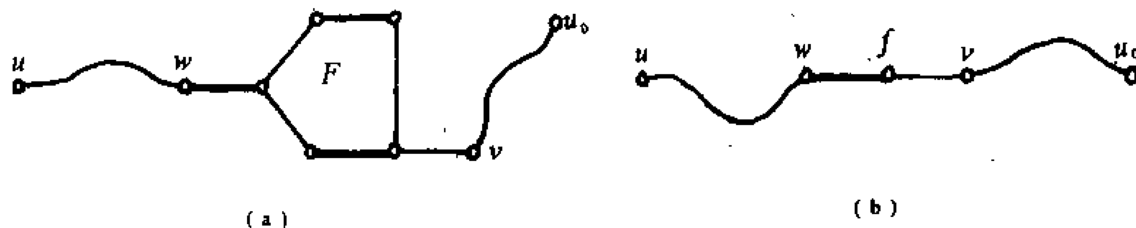


图 4.12 (a) G 中增广路 (b) G/F 中增广路

(2) P 以自由边进入 F,以匹配边离开 F,则类似于(1) 可得 G/F 中增广路.

(3) P 以自由边进入和离开 F. 由定理 4.10,对花蒂 v_0,必有 u-v_0 交错路 Q 且最后一条边是匹配边,此时,如图 4.13(a),

$$P = P' + P'' + P'''.$$

再细分为下面三种情形.

(3a) Q 与 P''' 相交,从 u 沿 Q 旅行最后离开 P' 或 P''' 的顶点 x 在 P''' 上,如图 4.13(b),x 分 Q 为 $Q_1 + Q_2$,分 P''' 为 $P_1 + P_2$,则 $P' + Q_2 + P_2$ 是 G/F 中的增广路.

(3b) Q 与 P''' 相交,从 u 沿 Q 旅行最后离开 P' 或 P''' 的顶点 x 在 P' 上,如图 4.13(c),x 分 Q 为 $Q_1 + Q_2$,分 P' 为 $P_1 + P_2$,则 $P_1 + Q_2 + P'''$ 是 G/F 中的增广路.

(3c) Q 与 P''' 不相交,如图 4.13(d),类似于(3b). 证毕.

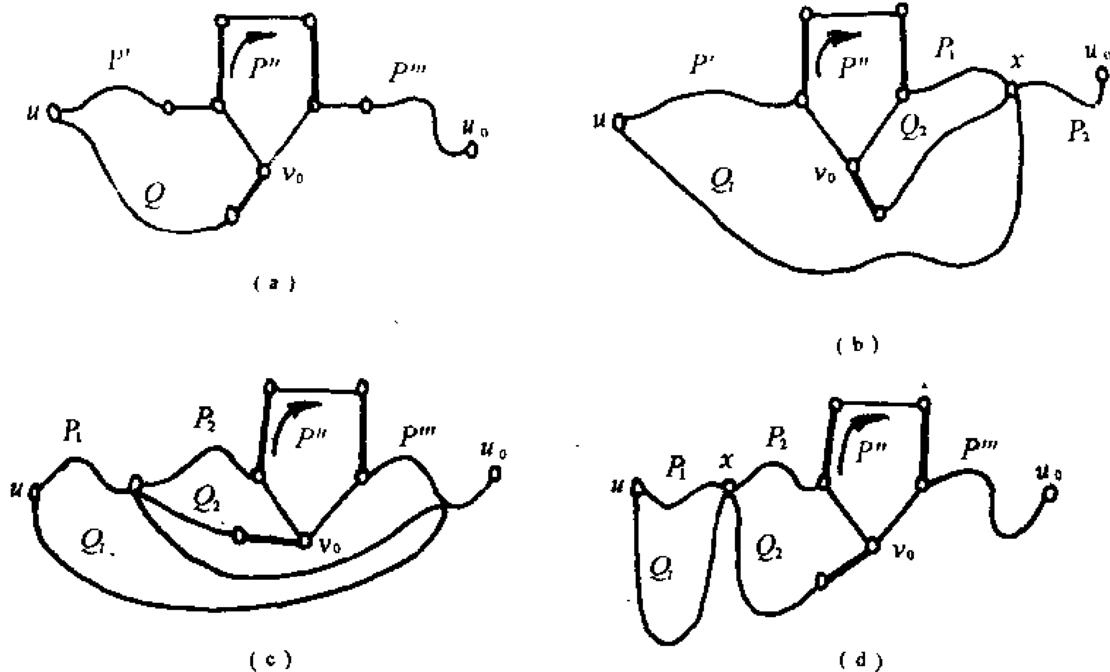


图 4.13 (a) 情形(3) (b) 情形(3a) (c) 情形(3b) (d) 情形(3c)

定理 4.11 和定理 4.2 指出,在 G 中求最大匹配等价于对花的收缩图 G/F 中求最大匹配. 对 G 而言,由于花的存在,使求增广路遇到障碍,对 G/F 而言,由于花被收缩,正好避开了这一障碍. 于是,求增广路的过程是,开始时在 G 中进行,一旦发现花就收缩,并在收缩图中继续进行(也许多次收缩). 当在收缩图中没有增广路时,原来的图 G 也没有增广路,则当前匹配就是最大匹配. 当在收缩图中找到增广路时,则打开花,按定理 4.11 证明中提

增广路，矛盾．故 Q 与 P 必相交．设 P 的两个 M-非饱和点是 u_0，v_0，Q 的两个 $M\oplus P$-非饱和点是 u，v．从顶点 u 出发沿 Q 旅行，与 P 的第一个公共顶点是 w，则 w 分 P 为两段，其中必有一段以 M 的边与 w 关联，如图 4.14 中的 u_0-w 段．于是，Q 的 u-w 段 $+$ P 的 w-u_0 段是从 u 出发的 M-增广路，矛盾．　证毕．

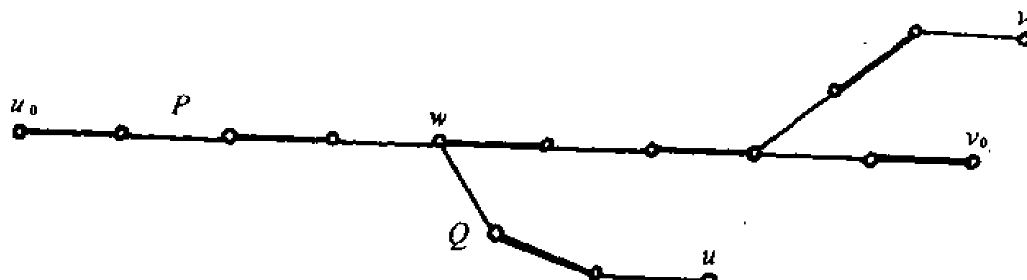


图 4.14　P 与 Q 被 $M\oplus P$ 匹配示意图

算法 4.5　花——求图的最大匹配的算法

设 M 是 $G=(V,E)$ 的匹配，u 是非饱和点，$\overline{G}$ 是 G 和 M 的导出有向图，则 $\overline{G}$ 中以 u 为根的外向树 $\overline{T}$ 对应 G 中以 u 为根的交错树 T．在 T 中若 u-v 路长是奇（偶）数，则称 v 为奇

$$V(T) = V(T) \cup \{j,k\}$$
$$E(T) = E(T) \cup \{e,e'\}$$
$$S = S \cup \{k\},$$

并继续生长交错树.

情形 3　$\exists\ i \in S, j \in S, e = (i,j) \notin E(T)$,如图 4.16(a).

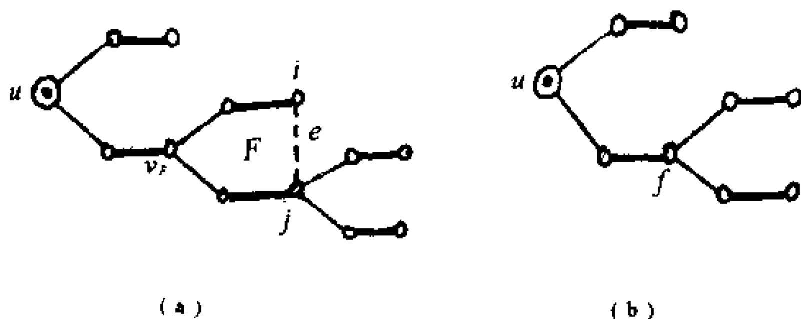


图 4.16　(a) 发现花 F　(b) 收缩花 F

此时得唯一圈 F. 设从 i 沿 $P(i)$ 逆向追踪与 $P(j)$ 的第一个公共顶点是 v_F,若 $v_F = u$,则 v_F 是偶点;若 $v_F \neq u$,则 $d_T(v_F) \geq 3$,而奇点的度为 2,故 v_F 也为偶点. 总之,v_F 是偶点. 故 F 是花,v_F 是花蒂,收缩花 F,如图 4.16(b). 在收缩图中,f 是偶点且其它顶点的奇偶性不变. 在收缩图中继续生长交错树.

情形 4　前三种情形都不出现,此交错树生长结束. 若不存在另外的非饱和点,当前匹配是最大匹配,否则选取另外一个非饱和点,开始另一个交错树的生长过程.

例 4.10　从图 4.7(a) 的图 G 和匹配 M,求 G 的最大匹配.

选非饱和点 v_1,按情形 2,得交错树 T,如图 4.17(a) 的实线部分,虚线为不在 T 的边.

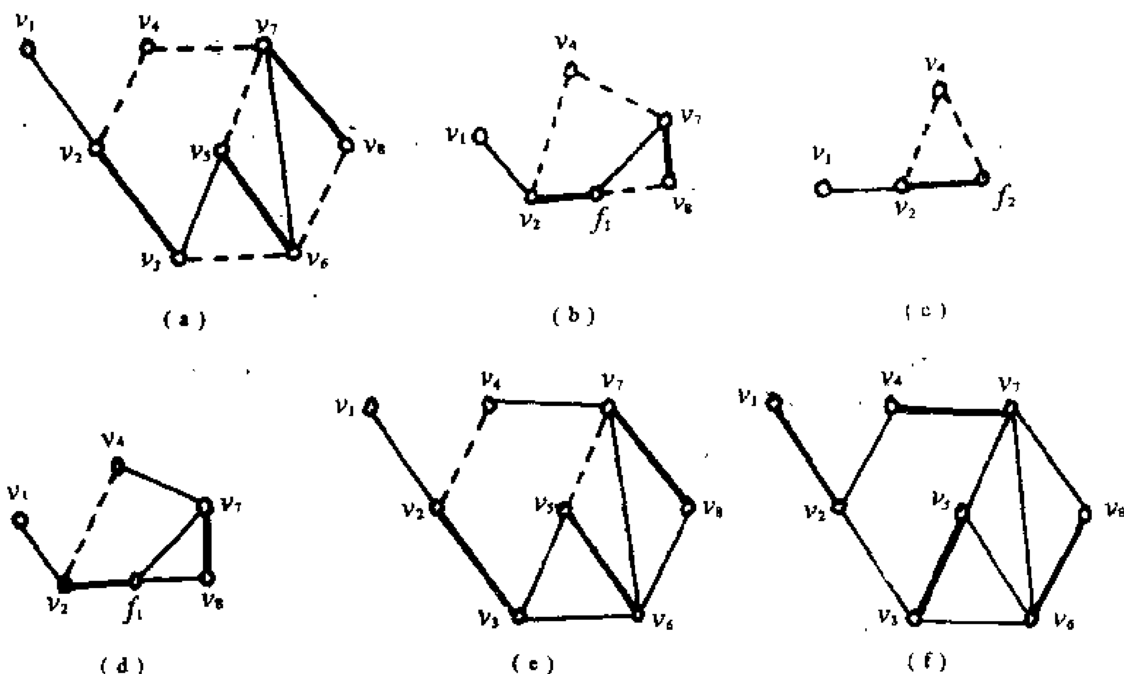


图 4.17　用算法 4.5 求最大匹配的过程

按情形 3，虚边(v_3,v_6)连接两个偶顶点，故得花 $F_1=[v_3,v_5,v_6]$. 收缩 F_1，如图 4.17(*b*)，令

$$G_1 \triangleq G/F_1,\quad T_1 \triangleq T/F_1.$$

按情形 3，虚边(f_1,v_8)连接两个偶顶点，故得花 $F_2=[f_1,v_8,v_7]$. 收缩 F_2，如图 4.17(*c*)，令

$$G_2 \triangleq G_1/F_2 \triangleq G/F_1/F_2,\quad T_2 \triangleq T_1/F_2 \triangleq T/F_1/F_2$$

按情形 1，在 G_2 有增广路

$$[v_1,v_2,f_2,v_4].$$

打开 f_2，并从顶点 v_4 逆向追踪，如图 4.17(*d*)，得 G_1 中的增广路

$$[v_1,v_2,f_1,v_8,v_7,v_4].$$

打开 f_1，并从顶点 v_6 逆向追踪，如图 4.17(*e*)，得 G 中的增广路

$$[v_1,v_2,v_3,v_5,v_6,v_8,v_7,v_4].$$

作对称差得新的匹配，如图 4.17(*f*). 因为没有非饱和点，所以已得最大匹配.

§4.8 应用——最优配对问题

在例 4.2 的配对问题中若再考虑工作效率，应如何配对使总的效率最大，这就是最优配对问题，即求赋权图的权最大的匹配问题.

容易看到，最大权匹配未必是最大匹配，而最大匹配可视为每条边的权都是 1 的最大权匹配.

注意到权为负的边不可能是最大权匹配中的边，因此不妨假设每条边的权非负.

设图 $G=(V,E)$ 是赋权图，$w(e)\geqslant 0\ \forall\ e\in E$. 匹配 M 的权是

$$w(M)=\sum\{w(e)\,|\,e\in M\}.$$

设 L 是交错路或交错圈，则称

$$p(L)=w(L\cap \overline{M})-w(L\cap M) \tag{4.15}$$

为判别数，这里 $\overline{M}=E\backslash M$.

定理 4.13 设 M 是赋权图 $G=(V,E)$ 的匹配，则 M 是最大权匹配的充要条件是对所有交错路和交错圈 L 都有 $p(L)\leqslant 0$.

证 必要性是明显的. 今证充分性. 设 M 是满足条件的匹配，M' 是任一个匹配. 作子图

$$H \triangleq (V,M\oplus M'),$$

由定理 4.1，H 的连通支是孤立点或交错路或交错圈，因此由条件导出 $w(M')\leqslant w(M)$. 证毕.

定理 4.14 设 $G=(V,E)$ 是赋权图，M 是有 k 条边的最大权匹配，L 是判别数最大的增广路，则 $M\oplus L$ 是有 $(k+1)$ 条边的最大权匹配.

证 令 $M'=M\oplus L$，则 $|M'|=k+1$. 由式(4.15)，

$$p(L)=w(M')-w(M).$$

若 M' 不是有 $(k+1)$ 条边的最大权匹配，则有 $(k+1)$ 条边的最大权匹配 M^*，使

$w(M^*) > w(M')$. 考虑子图

$$H \triangleq (V, M^* \oplus M)$$

的连通支.

(1) 有交错偶圈 C

若 $w(C \cap \overline{M}) - w(C \cap M) > 0$,则 $M \oplus C$ 是有 k 条边的匹配,且权大于 $w(M)$,矛盾.

若 $w(C \cap \overline{M}) - w(C \cap M) < 0$,则 $M^* \oplus C$ 是有$(k+1)$条边的匹配,且权大于 $w(M^*)$,矛盾.

所以 $w(C \cap \overline{M}) - w(C \cap M) = 0$.

(2) 有偶交错路 P,类似于(1)可得

$$w(P \cap \overline{M}) - w(P \cap M) = 0.$$

(3) 有奇交错路,其中两个端点都是 M-饱和点的设为 $P_1, P_2, \cdots, P_s$;两个端点都是 M-非饱和点的设为 $Q_1, Q_2, \cdots, Q_t$. 因为 $|M^*| = |M| + 1$,所以 $t = s + 1$. 若

$$w(P_i \cap \overline{M}) - w(P_i \cap M) + w(Q_i \cap \overline{M}) - w(Q_i \cap M) > 0,$$

则$(M \oplus P_i) \oplus Q_i$ 是有 k 条边的匹配,且权大于 $w(M)$,矛盾. 若

$$w(P_i \cap \overline{M}) - w(P_i \cap M) + w(Q_i \cap \overline{M}) - w(Q_i \cap M) < 0,$$

则$(M^* \oplus P_i) \oplus Q_i$ 是有$(k+1)$条边的匹配,且权大于 $w(M^*)$,矛盾. 所以

$$w(P_i \cap \overline{M}) - w(P_i \cap M) + w(Q_i \cap \overline{M}) - w(Q_i \cap M) = 0, \quad i = 1, 2, \cdots, s.$$

还剩一条 M-增广路 Q_t,有

$$p(Q_t) = w(M^*) - w(M) > w(M') - w(M) = p(L),$$

此与 L 的假设相矛盾. **证毕.**

下面用 S 表 V 的至少有 3 个顶点的奇基数顶点集,并称这样的 S 为奇集.

引理 4.15 对图 G 的任一匹配 M 和奇集 $S \subseteq V$,

$$|M \cap E(G[S])| \leqslant \frac{|S| - 1}{2}. \tag{4.16}$$

证 因为 $M \cap E(G[S])$ 是 $G[S]$ 的匹配,且 S 是奇集,故式(4.16)成立. **证毕.**

引理 4.16 设赋权图 $G = (V, E)$ 的每个顶点 i 有一个标号 $\pi(i) \geqslant 0$,每个奇集 S 有一个标号 $\mu(S) \geqslant 0$,若对每条边 $e = (i, j)$,满足

$$\pi(i) + \pi(j) + \sum_{\substack{S\text{奇集} \\ e \in E(G[S])}} \mu(S) \geqslant w(e), \tag{4.17}$$

则对 G 的每个匹配 M 有

$$w(M) \leqslant \sum_{i \in V} \pi(i) + \sum_{S\text{奇集}} \mu(S) \cdot \frac{|S| - 1}{2}. \tag{4.18}$$

证 式(4.17)对 M 的所有边求和,有

$$\sum_{e \in M} w(e) \leqslant \sum_{i} \pi(i) + \sum_{S} \mu(S), \tag{4.19}$$

其中右端第一个和式对所有 M-饱和点 i,第二个和式对所有奇集 S 且有匹配边在 $E(G[S])$ 中. 对这样的顶点集 S,$\mu(S)$ 至多出现 $\frac{|S| - 1}{2}$ 次,故从式(4.19)推出式(4.18).

证毕.

由引理 4.16 直接推出

定理 4.17 设 M 是赋权图 $G=(V,E)$ 的匹配，若存在 π,μ 满足式(4.17)，且

$$w(M)=\sum_{i\in V}\pi(i)+\sum_{S\text{奇集}}\mu(S)\cdot\frac{|S|-1}{2}, \tag{4.20}$$

则 M 是最大权匹配.

由此，找最大权匹配转化为如何构造标号 π,μ 的问题.

今对每条边 $e=(i,j)$ 定义标号 l：

$$l(e)=w(e)-\pi(i)-\pi(j)-\sum_{\substack{S\text{奇集}\\ e\in E(G[S])}}\mu(S), \tag{4.21}$$

则在下列条件下定理 4.17 的条件得到满足：

(Ⅰ) $l(e)\leqslant 0,\ e\in E$,

(Ⅱ) $l(e)=0,\ e\in M$,

(Ⅲ) $\pi(i)=0$, i 是 M- 非饱和点，

(Ⅳ) $\mu(S)=0$，当 $|M\cap E(G[S])|<\frac{|S|-1}{2}$. 换言之，若 $\mu(S)>0$ 则 $M\cap E(G[S])$ 是 $G[S]$ 的最大匹配.

算法的基本思想是构造一系列满足条件(Ⅰ)，(Ⅱ)和(Ⅳ)的匹配，使不满足(Ⅲ)的顶点数递减，最后达到全部四个条件都满足，从而求出最大权匹配.

算法 4.6 求最大权匹配的算法

设 $G=(V,E)$ 是赋权图，$w(e)\geqslant 0\ \forall\, e\in E$.

第 1 步 初始匹配 $M=\varnothing$，令

$\mu(S)=0$，当 S 为奇集，

$$\pi(i)=\frac{1}{2}\max_{j\in V}w(i,j),\ i\in V, \tag{4.22}$$

这样的选择自然满足条件(Ⅰ)，(Ⅱ)和(Ⅳ). 令

$$E_l=\{e\in E\,|\,l(e)=0\},$$

并称 $G_l=(V,E_l)$ 为**相等子图**.

第 2 步 若条件(Ⅲ)满足，则 M 是最大权匹配. 否则选择 M- 非饱和点 r 且 $\pi(r)>0$.

第 3 步 用 §4.7 的方法在 G_l 中构造以 r 为根的交错树 T. 在每一步，用 G_l 表 G_l 的收缩图. 若得到 G_l 中增广路 P，转第 4 步. 若对所有非饱和点都没有增广路，转第 5 步.

第 4 步 令 $M=M\oplus P$. 因为 $\pi(r)>0$，所以对新的匹配 M，使 $\pi(i)>0$ 的非饱和点至少减一个.

返回第 2 步.

第 5 步 若有偶点 j 使 $\pi(j)=0$，转第 6 步，否则转第 7 步.

第 6 步 存在 r 到 j 的偶交错路 Q，令

$$M=M\oplus Q,$$

因为 $\pi(j)=0,\pi(r)>0$，所以不满足(Ⅲ)的顶点数减少一个.

返回第 2 步.

第 7 步　修改 π,μ 如下：

$$\pi(i)=\begin{cases}\pi(i)-\varepsilon,\ i\text{ 是偶正常点或含于偶收缩点},\\ \pi(i)+\varepsilon,\ i\text{ 是奇正常点或含于奇收缩点}.\end{cases}\tag{4.23}$$

$$\mu(F_i)=\begin{cases}\mu(F_i)+2\varepsilon,\ i\text{ 是偶收缩点},\\ \mu(F_i)-2\varepsilon,\ i\text{ 是奇收缩点}.\end{cases}\tag{4.24}$$

其中正常点为图 G 的顶点，收缩点 i 是对应花 F_i 的点，并且 $\varepsilon>0$ 的选取应使

$$l(e)\leqslant 0\quad \forall\, e\in E,$$

$$\mu(S)\geqslant 0\quad \forall\ \text{奇集 } S,$$

$$\pi(i)\geqslant 0\quad \forall\, i\in V.$$

(1) 若 $e=(i,j)\in E\backslash E_l$，$i$ 是偶标号，j 未标号，则由式(4.23)，应有

$$w(e)\leqslant(\pi(i)-\varepsilon)+\pi(j)+\sum_{S}\mu(S),$$

即

$$\varepsilon\leqslant -l(e).\tag{4.25}$$

(2) 若 $e=(i,j)\in E\backslash E_l$，$i$ 和 j 都是偶标号，则由式(4.23)应有

$$w(e)\leqslant(\pi(i)-\varepsilon)+(\pi(j)-\varepsilon)+\sum_{S}\mu(S),$$

即

$$2\varepsilon\leqslant -l(e).\tag{4.26}$$

(3) 若 i 是偶正常点或含于偶收缩点，则由式(4.23)应有

$$\varepsilon\leqslant\pi(i).\tag{4.27}$$

(4) 若 i 是奇收缩点，则由式(4.24)应有

$$2\varepsilon\leqslant\mu(F_i).\tag{4.28}$$

因此，ε 的选取应是满足上面条件(1)～(4)的尽可能大的值. 当取定 ε 之后，有三种情形.

情形 1　若(1)或(2)取等号，则修改后的标号 $l(e)=0$，故这样的边都成为相等子图的边，返回第 3 步.

情形 2　若(3)取等号，则 $\pi(i)$ 变为零，且存在 r 到 j 的偶交错路，返回第 6 步.

情形 3　若(4)取等号，则 $\mu(F_i)$ 变为零，此时打开收缩点 i. 因为 i 是奇点，故在交错树 T 中关联于 i 的边只有两条，一条匹配边 e_2，一条自由边 e_1，如图 4.18(a).

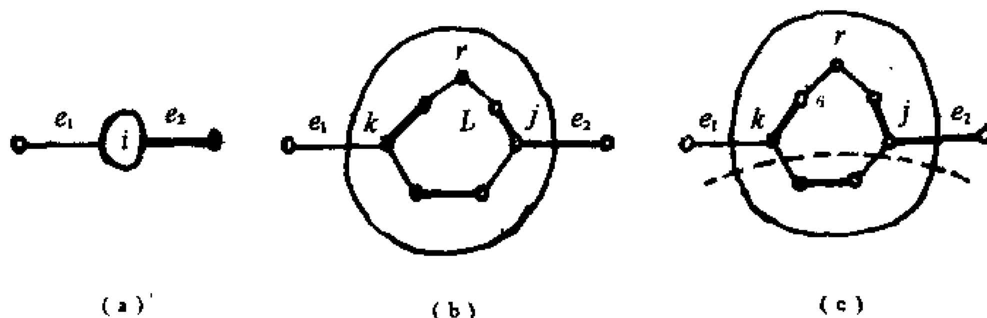


图 4.18　在情形 3 打开奇收缩点 i

注意到在打开 i 之前，算法第 4 步和第 6 步已经作过对称差而得到新的匹配，因此在

打开 i 之后应改变 F_i 中原来的匹配. 为此,设 e_2 与 F_i 中顶点 j 关联,e_1 与 F_i 中顶点 k 关联,从顶点 j 到花蒂 r 的偶交错路为 L,如图 4.18(b).

作对称差 $M \oplus L$,如图 4.18(c). 并删去 F_i 中不经过顶点 r 的 k-j 交错路,如图4.18(c)中虚线标明的交错路,由此即得打开奇收缩点 i 后的交错树. 返回第 3 步.

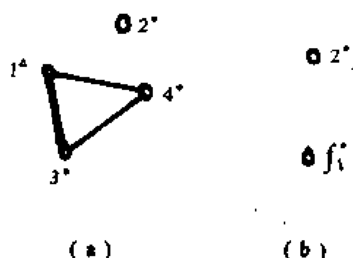
例 4.11 给出下列赋权完全图

顶点	1	2	3	4
1	0	3	5	5
2	3	0	2	3
3	5	2	0	5
4	5	3	5	0

试求最大权匹配.

开始设 $M = \varnothing$. 以 * 表偶点,△ 表奇点,未标者不在当前交错树中.

由式(4.22) 得初始顶点标号,如表 4.4.



(a) (b)

图 4.19

表 4.4

i	$\pi(i)$
1	2.5
2	1.5
3	2.5
4	2.5

作相等子图 G_l 如图 4.19(a);G_l 的最大匹配是

$$M = \{(1,3)\},$$

收缩花 $F_1 = [1,3,4]$,如图 4.19(b)

修改标号如下:

由条件(4.26), $2\varepsilon \leqslant -l(2,4) = \pi(2) + \pi(4) - w(2,4) = 1$,

$$2\varepsilon \leqslant -l(2,3) = \pi(2) + \pi(3) - w(2,3) = 2.$$

由条件(4.27), $\varepsilon \leqslant \min\{\pi(1),\pi(2),\pi(3),\pi(4)\} = 1.5$,

所以 $\varepsilon = 0.5$. 按式(4.23) 和式(4.24) 得顶点与奇集 F_1 的标号如表 4.5.

调整标号后 $l(1,2) = l(2,4) = 0$,故边(1,2)与(2,4) 加入相等子图中.

表 4.5

i	$\pi(i)$	$\mu(F_1) = 1$
1	2	
2	1	
3	2	
4	2	

在收缩图中得最大匹配,如图 4.20(a).

打开 f_1 如图 4.20(b) 或(c). 由(b),最大权匹配是

$$\{(1,3),(2,4)\}.$$

由(c),在 F_1 中顶点 1 到花蒂 4 的交错路,经对称差

得图 4.20(d)，故得最大权匹配

$$\{(1,2),(3,4)\}.$$

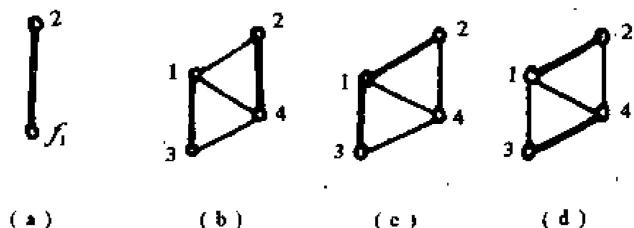


图 4.20

习　题　四

4.1　设 M 是图 G 的匹配，若 M- 非饱和点至多一个，则 M 是最大匹配.

4.2　图 $G=(V,E)$ 的任意两个最大匹配都能用对称差从一个得到另一个.

4.3　给定图 G，两个人做游戏，在图 G 上依次交错选点，其规则是

(1) 第一次选点任意，

(2) 从第二次选点开始，每次选点必须与紧前选点相邻，

(3) 当无法选点时终止，

(4) 选得最后一点者为胜，

试证先选者必胜的充要条件是 G 没有完美匹配.

4.4　在图 4.21 中给定图 G 的匹配 M(M 的边用粗线表示)，试由定理 4.2 求最大匹配.

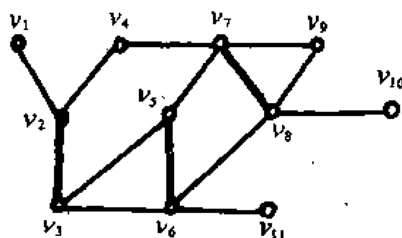


图 4.21　图 G 的匹配 M

4.5　证明没有桥的 3- 正则图有完美匹配.

4.6　习题 4.5 的结论若去掉条件“没有桥”则不成立，试举例说明.

4.7　对任意正整数 $k>1$，存在没有完美匹配的 k 正则图.

4.8　设 T 是树，则 T 有完美匹配的充要条件是

$$\omega_0(T-v)=1,\ v\in V(T).$$

4.9　证明树至多一个完美匹配.

4.10　证明 k- 立方有完美匹配.

4.11　设 G 是$(k-1)$- 边连通的 k- 正则图，且 $|V(G)|$ 是偶数，则 G 有完美匹配.

4.12　(n,m) 图 G 的最大匹配的边数是$\frac{n-d}{2}$，其中

$$d=\max_{S\subseteq V}\{\omega_0(G-S)-|S|\}.$$

4.13　k- 正则偶图($k>0$) 有完美匹配.

4.14　举例说明定理 4.7 中的条件“偶图”不能去掉.

4.15　设 $A_1,A_2,\cdots,A_m$ 是 S 的子集，如果每个子集 A_i 任选一元 $a_i\in A_i$，使 $a_1,\cdots,a_m$ 互不相同，则称$\{a_1,\cdots,a_m\}$ 为子集簇$\{A_1,\cdots,A_m\}$ 的一个代表系. 证明$\{A_1,\cdots,A_m\}$ 有代表系的

充要条件是对任意的 $J \subseteq \{1,2,\cdots,m\}$ 都有

$$\left|\bigcup_{j\in J} A_j\right| \geqslant |J|.$$

4.16　设 A 是 0-1 矩阵，则划去所有零元素所需横竖线的最少条数，等于不同行不同列的零元素的最大数目.

4.17　设 $G=(X,Y,E)$ 是偶图，则 G 的最大匹配的边数是

$$|X| - \max_{S\subseteq X}\{|S| - |\Gamma(S)|\}.$$

4.18　设 $G=(X,Y,E)$ 是简单偶图，$|X|=|Y|=n$，$|E|>(k-1)n$，则 G 有边数为 k 的匹配.

4.19　设 $G=(X,Y,E)$ 是偶图，$V=X\cup Y$，M_1 和 M_2 是二个匹配，则存在匹配 $M\subseteq M_1\cup M_2$，使 M 饱和所有由 M_1 饱和的 X 的顶点和所有由 M_2 饱和的 Y 的顶点.

4.20　设 M 是图 G 的匹配，则存在最大匹配 M^* 使 M^* 饱和所有 M 饱和点.

4.21　设 $d_G(v)>0$，则必有最大匹配饱和 v.

4.22　分配六人作六件工作如下表，“1”表胜任，“0”表不胜任，能否使每个人都作他能胜任的工作？

工作 人员	B_1	B_2	B_3	B_4	B_5	B_6
A_1	1	0	1	0	0	0
A_2	0	1	1	1	0	0
A_3	1	0	0	1	0	0
A_4	0	1	0	1	1	0
A_5	0	0	0	1	0	1
A_6	0	0	0	1	1	1

4.23　有七人去应聘五件工作如表所示，“1”表示愿意应聘，“0”表示不愿意应聘. 问最多能有多少人被应聘？设初始匹配为 $\varnothing$.

人员 工作	A_1	A_2	A_3	A_4	A_5	A_6	A_7
B_1	1	1	0	1	0	0	0
B_2	1	0	1	1	1	1	0
B_3	0	1	1	1	0	1	1
B_4	0	0	0	0	1	0	0
B_5	0	0	0	0	1	0	0

4.24　设 $G=(X,Y,E)$ 是偶图，若存在 $t\geqslant 1$ 使

(1) $d(x)\geqslant t,\ x\in X$;

(2) $d(y)\leqslant t,\ y\in Y$;

则 G 有匹配饱和 X 的所有顶点.

4.25　设 $G=(X,Y,E)$ 是偶图，若 $E\neq\varnothing$，对任意 $x\in X$ 和 $y\in Y$ 都有 $d(x)\geqslant d(y)$，

则有匹配饱和 X 的所有顶点.

4.26　设图 $G=(V,E)$，$N\subseteq V$，若 N 的任意两个顶点都不相邻，则称 N 为**独立集**. 试证：N 是独立集的充要条件是 $V\backslash N$ 是 G 的覆盖.

4.27　图 G 的最小覆盖的顶点数称为**覆盖数**，记为 α_0，最大独立集的顶点数称为**独立数**，记为 β_0，试证

$$\alpha_0+\beta_0=|V(G)|.$$

4.28　设图 $G=(V,E)$，$S\subseteq E$，若每个顶点都与 S 的一条边关联，则称 S 是一个**边覆盖**. 边数最少的边覆盖称为**最小边覆盖**，其元素个数称为**边覆盖数**，记为 α_1. 最大匹配所含边数称为**匹配数**，记为 β_1. 试证

$$\beta_1\leqslant\alpha_0,\ \beta_0\leqslant\alpha_1.$$

4.29　设图 $G=(V,E)$ 没有孤立点，则

$$\alpha_1+\beta_1=|V|.$$

4.30　设 $G=(V,E)$ 没有孤立点，问如何从最大匹配构造最小边覆盖？如何从最小边覆盖求最大匹配？

4.31　设 G 是没有孤立点的偶图，则

$$\alpha_0=\alpha_1,\ \beta_0=\beta_1.$$

4.32　设 S 是图 G 的边覆盖，则 S 是最小边覆盖的充要条件是，S 不含长大于等于 3 的路.

4.33　设图 $G=(V,E)$，$S\subseteq V$，若 $G[S]$ 是完全图，则称 S 是 G 的一个**团**. 有 k 个顶点的团称为 k **团**. 有 l 个顶点的独立集称为 l **独立集**. 拉姆瑟数 $r(k,l)$ 是具有下述性质的最小正整数：

(1) $r(1,l)=r(k,1)=1$；

(2) $r(2,l)=l,\ r(k,2)=k,\ (k,l\geqslant 2)$；

(3) 当 $k,l\geqslant 2$，若 $r(k,l-1)$ 与 $r(k-1,l)$ 存在，则 $r(k,l)$ 也存在且

$$r(k,l)\leqslant r(k,l-1)+r(k-1,l).$$

当 $r(k,l-1)$ 与 $r(k-1,l)$ 都是整数时，上式的不等号是严格的. 试证任一至少有 $r(k,l)$ 个顶点的简单图，或者含有 k 团，或者含有 l 独立集.

4.34　证明

$$r(k,l)\leqslant\binom{k+l-2}{k-1}.$$

4.35　在习题 1.12 中的几个命题实际上是拉姆瑟数问题. 试证：

$$r(3,3)=6,r(3,4)=9,r(3,5)=14,r(4,4)=18.$$

4.36　证明 $r(k,l)=r(l,k)$.

4.37　设 G 是不含三角形的简单图，则

$$\Delta\leqslant\beta_0.$$

4.38　当 $k\geqslant 3$ 时，$r(k,k)>k2^{\frac{k}{2}-2}$.

4.39　证明 $r(k,l)>m^{\frac{m}{2}-2}$，这里 $m=\min\{k,l\}\geqslant 2$.

4.40　用可行顶点标号法求最优分配，已知

w_{ij}	y_1	y_2	y_3	y_4	y_5
x_1	3	5	5	2	4
x_2	1	4	2	1	1
x_3	2	3	3	0	5
x_4	1	4	0	3	2
x_5	3	4	6	5	0

4.41　(1) 如何用可行顶点标号法求赋权完全偶图的最小权完美匹配？

(2) 用(1)的方法求最小权完美匹配，已知

w_{ij}	y_1	y_2	y_3	y_4	y_5
x_1	4	5	8	10	11
x_2	7	6	5	7	4
x_3	8	5	12	9	6
x_4	6	6	13	10	7
x_5	4	5	7	9	8

4.42　如图 4.22，由算法 4.4 求最大匹配.

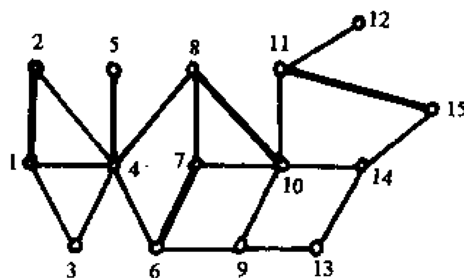


图 4.22

4.43　利用算法 4.5 求图 4.22 的最大匹配.

4.44　利用算法 4.6 求最大权匹配，已知

顶点	1	2	3	4	5	6	7	8
1		18	29	29	19	19	12	8
2			37	29	27	33	22	14
3				33	29	35	22	19
4					35	27	22	21
5						27	15	12
6							18	18
7								0
8								

4.45　利用算法 4.6 求最大权匹配，已知

顶点	1	2	3	4	5
1	╲	3	5	4	2
2		╲	2	1	4
3			╲	6	3
4				╲	6
5					╲

4.46　(1) 试修改算法 4.6 以求赋权完全图的最小权匹配.

(2) 试用(1)的算法求最小权匹配，已知

顶点	1	2	3	4	5	6	7	8
1	╲	20	1	5	24	9	11	16
2		╲	9	19	30	9	19	26
3			╲	3	19	5	9	16
4				╲	20	11	11	20
5					╲	17	26	28
6						╲	3	16
7							╲	23
8								╲

第五章　平面图及其应用

§5.1　平面图和可平面图

例 5.1　相传有个国王临终把国土分成五块，分给五个儿子.五个儿子在各自的领地建一王宫，要求各王宫有路直通而不交叉.这五个儿子的要求能办到吗？

若把王宫作为顶点，王宫之间的直通路当作边，则问题是：五个顶点的完全图 K_5 能画在平面上，使各边在不是顶点处都不相交吗？

例 5.2　一个古老的问题，据说有相距不远的三户人家和三口井，欲修路使三户人家各自与三口井直通，而这些路互不交叉，能办到吗？

把三户人与三口井当作顶点，三户人家各自到三口井的直通路作为边，则问题是：完全偶图 $K_{3,3}$ 能画在平面上，使各边在不是顶点处都不相交吗？

例 5.3　在现代，印刷电路的使用是很普遍的.一块印刷电路板可视为画在平面上的一个图.面常常遇到的问题是如何设计印刷电路，用图论的语言，即怎样的图才能画在一张平面上使各边在不是顶点处都不相交？

定义 5.1　若能把一个图 G 的图形画在曲面 S 上，使图的边在顶点之外都不相交，则称图 G **可嵌入曲面** S.可嵌入平面的图称为**可平面图**，否则称为**不可平面图**.已经嵌入一张平面的图称为**平面图**.

从定义即知，可平面图就是同构于一个平面图的图.

例 5.4　图 5.1(a) 的图 G 是可平面图，因为它可以画在平面上如图 5.1(b)，或者说 (a) 的图同构于(b) 的平面图.

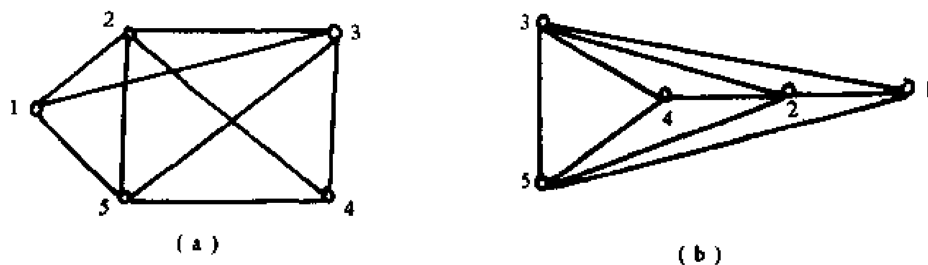


图 5.1　可平面图和平面图

定义 5.2　设 G 是平面图，G 的一个**面**是由 G 的边所围成的一个封闭区域，该区域内的任二点都有一条曲线相连接，且此曲线不遇 G 的任何边和顶点.界定一个面的所有边称为该面的**边界**，边界中的边数称为该面的**度**，并规定桥在计算度时算作两条边.对于面 f 的边界上的一条边 e，也称 e 是 f 的一条边界.面 f 的**周线**是由面 f 的边界构成且把面 f 包

含在内的圈. 两个面若有公共边则称它们相邻. 一个面与它的边界上的边和点称为**相关联**.

每个平面图恰有一个无界面,称为**外部面**,其余的面都是有界面,称为**内部面**.

由定义,外部面没有周线,内部面都有一条周线.

例5.5　图5.2的平面图有四个面:f_1,f_2,f_3和f_4. 其中f_4是外部面,其余为内部面. 各面的度、边界和周线如表5.1.

表5.1

面	度	边　　界	周　　线
f_1	6	$e_1e_3e_3e_4e_5e_2$	$e_1e_4e_5e_2$
f_2	2	e_7e_8	e_7e_8
f_3	1	e_{10}	e_{10}
f_4	11	$e_1e_4e_6e_7e_9e_{10}e_9e_8e_6e_5e_2$	没有周线

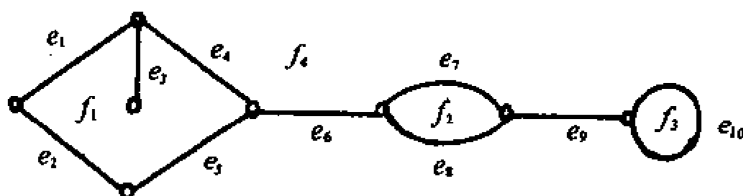


图5.2

定理5.1　设G是平面图,则

$$\sum_{f\in F(G)} d(f) = 2|E(G)|, \tag{5.1}$$

其中$F(G)$表G的所有面的集,$d(f)$表面f的度.

证　不是桥的边正好在两个面的公共边界上,面桥在同一个面内按规定计算两次,因此在计算面的度时,每条边刚好重复一次,故式(5.1)成立.　**证毕**.

式(5.1)类似于第一章的式(1.1). 并如同推论1.2,从定理5.1直接导出下面的推论.

推论5.2　平面图中奇度面数必为偶数.

定理5.3　设G是平面图,则G的所有内部面的周线构成圈基.

证　若G没有内部面,定理自然成立.

假设对具有$(k-1)(k\geqslant 1)$个内部面的平面图定理成立. 今设G是有k个内部面的平面图. 若这些内部面互不相邻,其周线自然构成圈基. 若内部面f_1与f_2有公共边e,则$G'\triangleq G-e$是有$(k-1)$个内部面的平面图. 由归纳假设,G'的周线构成圈基. 现在在G'中回复边e得G,则由f_1与f_2的二条含边e的周线代替了在G'的不含边e的一条周线,从而得到G的k条周线. 容易看到G的这k条周线是独立的,并且,G的任一圈都能由其线性表示,因此G有k条周线构成的圈基.　**证毕**.

定理5.4　平面图可以嵌入平面使任一个有界面成为无界面.

证　设 f 是一个有界面. 现将平面图嵌入球面,选 f 内的一个点作北极,并作平面切于南极,然后向该切平面作球极投影,则该平面图已嵌入平面且 f 成为外部面.　**证毕.**

§ 5.2　Euler 公式

定理 5.5　设 G 是连通平面图,则

$$|V(G)| - |E(G)| + |F(G)| = 2. \tag{5.2}$$

证　由定理 5.3,G 的 $(|F(G)| - 1)$ 个内部面的周线构成圈基,即 $\eta(G) = |F(G)| - 1$,再由第三章式(3.5)即得式(5.2).　**证毕.**

式(5.2)称为 Euler 公式,它是数学的经典公式之一.

推论 5.6　设 G 是平面图,则

$$|V(G)| - |E(G)| + |F(G)| = \omega(G) + 1. \tag{5.3}$$

证　设 G 的连通支是 $G_1, G_2, \cdots, G_\omega$,则

$$|V(G_i)| - |E(G_i)| + |F(G_i)| = 2,\ i = 1, 2, \cdots, \omega,$$

上面 ω 个等式相加得

$$|V(G)| - |E(G)| + |F(G)| + (\omega - 1) = 2\omega,$$

整理即得式(5.3).　**证毕.**

定理 5.7　设 G 是简单连通平面图,$|V(G)| \geqslant 3$,每个面的边界至少含 k 条边,则

$$\frac{k}{2}|F(G)| \leqslant |E(G)| \leqslant \frac{k}{k-2}(|V(G)| - 2). \tag{5.4}$$

证　作偶图 G':顶点集 X 对应 G 的面,顶点集 Y 对应 G 的边;$x^* \in X$ 与 $y^* \in Y$ 相邻,当且仅当面 x 与边 y 关联,则

$$k|F(G)| \leqslant |E(G')| \leqslant \sum_{f \in F(G)} d(f).$$

把式(5.1)和 Euler 公式代入上式即得式(5.4).　**证毕.**

推论 5.8　K_5 是不可平面图,即例 5.1 的问题的答案是否定的.

证　K_5 的每个面至少含 3 条边. 将 $k = 3$, $|E(K_5)| = 10$, $|V(K_5)| = 5$ 代入式(5.4)的右边不等式得 $10 \leqslant 9$,矛盾.　**证毕.**

推论 5.9　$K_{3,3}$ 是不可平面图,即例 5.2 的问题的答案是否定的.

证　$K_{3,3}$ 不含奇圈,故 $K_{3,3}$ 的每个面至少有 4 条边,将 $k = 4$, $|E(K_{3,3})| = 9$, $|V(K_{3,3})| = 6$ 代入式(5.4)的右边不等式得 $9 \leqslant 8$,矛盾.　**证毕.**

定理 5.10　设 G 是简单平面图,$|V(G)| \geqslant 3$,则

$$|E(G)| \leqslant 3|V(G)| - 6, \tag{5.5}$$

(习题 5.3).

定义 5.3　称 G 是极大可平面图,如果 G 是简单可平面图,且 G 的任二不相邻顶点附加一条边即成为不可平面图.

例 5.6　K_5 任去一边的图如图 5.1,是可平面的;又由推论 5.8,K_5 是不可平面的,所以 K_5 任去一边的图是极大可平面图.

定理 5.11　设 G 是极大可平面图,$|V(G)| \geqslant 3$,则 G 的每个面都是三角形且

$$|E(G)| = 3|V(G)| - 6, \tag{5.6}$$

$$|F(G)| = 2|V(G)| - 4, \tag{5.7}$$

(习题 5.7).

定义 5.4 设 G 是简单连通图,若能把 G 嵌入平面使 G 的所有顶点在同一个面上,则称 G 是**外可平面的**. 若 G 是外可平面的,且任二不相邻顶点附加一条边即成为不外可平面的,则称 G 是**极大外可平面的**.

定理 5.12 设 G 是外可平面图, $|V(G)| \geqslant 2$,则

$$|E(G)| \leqslant 2|V(G)| - 3. \tag{5.8}$$

证 若 G 没有有界面,则 G 是树,故

$$|E(G)| = |V(G)| - 1,$$

式(5.8) 自然成立. 若 G 有有界面,则 $d(f) \geqslant 3$,故

$$2|E(G)| = \sum_{f\in F(G)} d(f) \geqslant 3(|F(G)| - 1) + |V(G)|, \tag{5.9}$$

再由 Euler 公式, $|F(G)| = |E(G)| - |V(G)| + 2$,代入式(5.9) 即得式(5.8). **证毕.**

定理 5.13 设 G 是极大外可平面图, $|V(G)| \geqslant 2$,则 G 的面除一个外都是三角形,且

$$|E(G)| = 2|V(G)| - 3, \tag{5.10}$$

$$|F(G)| = |V(G)| - 1. \tag{5.11}$$

证 若 G 没有有界面,则

$$|V(G)| = 2,\ |E(G)| = 1,\ |F(G)| = 1,$$

式(5.10) 与(5.11) 自然成立. 若有有界面,则 $d(f) \geqslant 3$. 再由极大性, $d(f) = 3$. 即所有内部面都是三角形. 此时,式(5.9) 成为等式,再由 Euler 公式,即得式(5.10) 与(5.11). **证毕.**

由定理 5.11,极大平面图是球面的一个三角剖分. 由定理 5.13,极大外平面图是多边形的一个三角剖分.

例 5.7 有 6 个顶点的极大外平面图,是六边形的三角剖分,由此可得 6 个顶点的所有极大外平面图,如图 5.3.

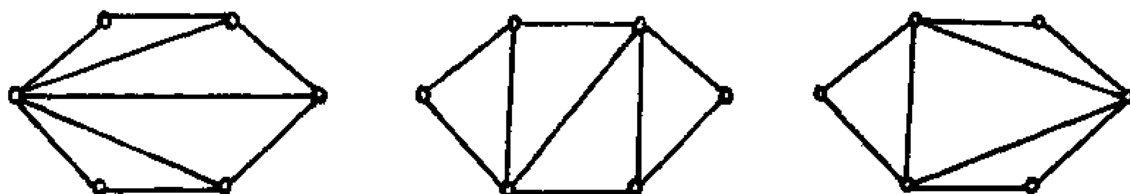


图 5.3 6 个顶点的所有极大外平面图

§ 5.3 Kuratowski 定理

研究平面图的一个重要问题就是如何判定一个图是不是可平面的. 由于图的连通支是图的极大连通子图,所以一个图可平面,当且仅当各连通支都可平面. 由于连通图的块 - 分离点图是树,所以一个图可平面,当且仅当各块都可平面. 但是,块的可平面性如何判定呢?这里并未解决. 关于图的可平面性判定,最早是由 Kuratowski 给出的[47].

定义 5.5 在图 G 的边 (u,v) 上添加 k 个顶点 $v_1,v_2,\cdots,v_k(k\geqslant 1)$，而使边 (u,v) 变为 $(k+1)$ 条边

$$(u,v_1),(v_1,v_2),\cdots,(v_k,v),$$

则称为对边 (u,v) 的加细. 两个图称为**同胚的**，如果其中一个图是另一个图的加细图.

容易看到，图 G 可平面，当且仅当 G 的加细图是可平面的. 由于 K_5 与 $K_{3,3}$ 不可平面，因此，K_5 与 $K_{3,3}$ 的加细图也是不可平面的. 换言之，一个图若有同胚于 K_5 或 $K_{3,3}$ 的子图，则该图不可平面. 反过来，不可平面图是否一定包含同胚于 K_5 或 $K_{3,3}$ 的子图?Kuratowski 作出了肯定的回答.

定理 5.14(Kuratowski 定理) 一个图是可平面的充要条件是 G 没有同胚于 K_5 或 $K_{3,3}$ 的子图.

证 必要性已证，今证充分性. 用反证法. 设 G 是具有下列性质的边数最少的图：不可平面，且没有同胚于 K_5 或 $K_{3,3}$ 的子图. 先证明若干结论是方便的.

(1) 由边数最少推出 G 是一个块，$\delta(G)\geqslant 3$，且任去 G 的一边都变为可平面的.

(2) 设 $e_0=(u_0,v_0)\in E(G)$，则 $G-e_0$ 有圈含 u_0 与 v_0. 不然，由定理 2.10，u_0 与 v_0 在 $G-e_0$ 的不同块中，故 $G-e_0$ 有分离点 w，w 在每一条 u_0-v_0 路上. 如果 $G-e_0$ 中没有边 (u_0,w) 和 (w,v_0)，则添加它们构成图 H，且 u_0,v_0 仍在 H 的不同块中，例如 B_1 和 B_2. 因为 B_1 和 B_2 的边都少于 G 的边，所以 B_1 或者可平面，或者有同胚于 K_5 或 $K_{3,3}$ 的子图. 如果 B_1 有子图同胚于 K_5 或 $K_{3,3}$，则用 $G-e_0$ 中的 u_0-w 路代替 (u_0,w) 得到 G 的子图同胚于 K_5 或 $K_{3,3}$，矛盾. 因此 B_1 可平面. 同理 B_2 可平面. 由定理 5.4，B_1 和 B_2 都可以画在平面上使 (u_0,w) 和 (w,v_0) 在外部面的边界上. 于是，在 H 添加边 e_0 仍是可平面的，再去掉 (u_0,w) 和 (w,v_0) 即得 G，从而 G 也可平面，矛盾. 故结论(2) 成立.

把 $G-e_0$ 嵌入平面，并设 C 是含 u_0 与 v_0 的圈中含有面数最多的一个圈. 指定 C 的一个方向，并用 $C[u,v]$ 表示在圈 C 上沿指定方向从 u 到 v 的路. 当 v 在 C 上不是 u 的紧后顶点时，用 $C(u,v)$ 表示 $C[u,v]$ 去掉 u 和 v 之后的子路. 在 C 外的顶点的导出子图称为 C 的**外部**，外部的连通支称为**外部支**. 两个端点都在某一个外部支以及一个端点在该外部支而另一个端点在 C 上的所有边导出子图称为一个**外片**. 类似可定义 C 的**内部**、**内部支**和**内片**. 外片或内片若含有 C 上的顶点则称为与 C **接触**，而该顶点称为**接触点**. 一个外片或内片若与 $C(u,v)$ 和 $C(v,u)$ 都接触，则称为**分离** u-v.

(3) 每一个外片恰有 2 个接触点且分离 u_0-v_0.

因为 $G-e_0$ 连通，且没有分离点，所以每个外片至少有 2 个接触点. 一个外片不能与 $C(u_0,v_0)$ 或 $G(v_0,u_0)$ 有多于一个的接触点，不然，将有一个含 u_0 和 v_0 的圈而包含的面数更多，矛盾. 故结论(3) 得证.

(4) 存在内片分离 u_0-v_0. 不然，所有内片都与边 e_0 不相交，故 $(G-e_0)+e_0$ 即 G 可平面，矛盾.

(5) 存在外片分离 u_0-v_0，它与 $C(u_0,v_0)$ 在 u_1 接触，与 $C(v_0,u_0)$ 在 v_1 接触，使得存在内片分离 u_0-v_0，也分离 u_1-v_1. 不然，将分离 u_0-v_0 的内片编号：从 u_0 沿 C 第一个遇到内片的接触点时，记该内片为 I_1，依次记为 I_2,I_3 等等. 设 I_1 接触 $C(u_0,v_0)$ 的第一个与最后一个顶点是 u_2，u_3(也许重合)，接触 $C(v_0,u_0)$ 的第一个与最后一个顶点是 v_2,v_3(也许重合). 如图 5.4 所示.

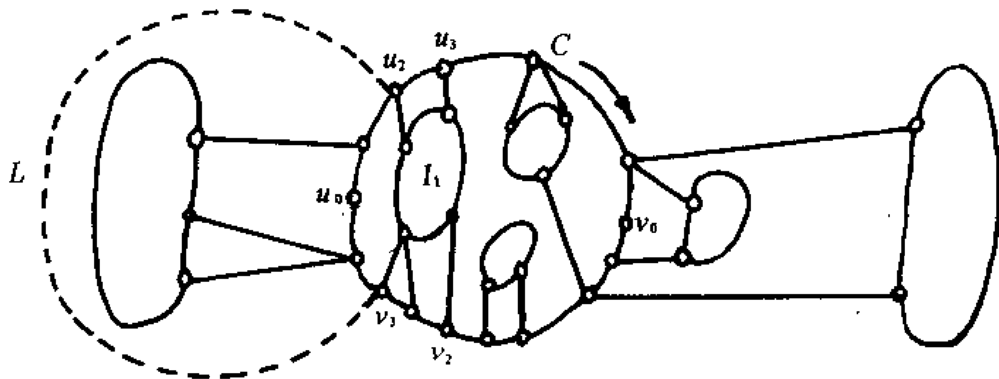


图 5.4

由于在 $C(v_3,u_2)$ 没有内片接触点，所以，在结论(5) 不成立时，外片的 2 个接触点或者都在 $C[v_3,u_2]$ 上或者都在 $C[u_3,v_2]$ 上. 今在外部区域画一条连接 v_3 与 u_2 的曲线 L(图 5.4 中虚线所示)，除顶点 v_3 与 u_2，L 不与 $G-e_0$ 相交. 于是，I_1 可以保持平面性而移到 L 的外面. 类似地，可依次将其余所有分离 u_0-v_0 的内片保持平面性而移到圈 C 的外面. 这样一来，将边 $e_0=(u_0,v_0)$ 加入 $G-e_0$ 仍保持平面性，此与 G 不可平面相矛盾. 所以结构(5) 成立.

现在回到定理的证明. 设 F 是合于结论(5) 的一个内片，即 F 分离 u_0-v_0，也分离 u_1-v_1. 设 F 分别与 $C(u_0,v_0)$，$C(v_0,u_0)$，$C(u_1,v_1)$ 与 $C(v_1,u_1)$ 的接触点是 w_0,y_0,w_1 与 y_1，如图 5.5(a)，则有下列五种情形.

情形 1　w_1 与 y_1 一个在 $C(u_0,v_0)$，另一个在 $C(v_0,u_0)$，如图 5.5(b)，其中 v_1-u_1 路是通过外片的，y_1-w_1 路是通过内片的. 此时 G 含有同胚于 $K_{3,3}$ 的子图. 图中分别用空心点与实心点表示偶图顶点的分解.

情形 2　w_1 与 y_1 都在 $C(u_0,v_0)$ 或 $C(v_0,u_0)$ 上，不妨设是前者. 又有三种可能.

(2a)　y_0 在 $C(u_1,v_1)$ 上，如图 5.5(c)；

(2b)　y_0 在 $C(v_1,u_1)$ 上，如图 5.5(d)；

(2c)　$y_0=v_1$，如图 5.5(e)，在内片 F 上有一顶点 r，存在 r-y_1 路，r-w_1 路和 r-v_1 路.

以上三种可能，G 都有子图同胚于 $K_{3,3}$，图中虚线为不在 $K_{3,3}$ 的部分.

情形 3　$w_1=v_0$，$y_1\neq u_0$，y_1 在 $C(u_0,v_0)$ 或 $C(v_0,u_0)$ 上，不妨设是前者. 又有三种可能.

(3a)　y_0 在 $C(u_1,v_1)$ 上，如图 5.5(f)；

(3b)　y_0 在 $C(v_1,u_1)$ 上，如图 5.5(g)；

(3c)　$y_0=v_1$，如图 5.5(h)，在内片 F 上有一顶点 r，存在 r-y_1 路，r-v_0 路与 r-v_1 路.

以上三种可能，G 都有子图同胚于 $K_{3,3}$.

情形 4　$w_1=v_0$，$y_1=u_0$，w_0 在 $C(v_1,u_1)$ 或 $C(u_1,v_1)$ 上，不妨设是前者. 又有三种可能.

(4a)　y_0 在 $C(u_1,v_1)$ 上，如图 5.5(i)；

(4b)　y_0 在 $C(v_1,u_1)$ 上，如图 5.5(j)；

(4c)　$y_0=v_1$，如图 5.5(k).

类似于上面的分析，G 有子图同胚于 $K_{3,3}$.

情形 5　$w_1=v_0$，$y_1=u_0$，$w_0=u_1$，$y_0=v_1$，如图 5.5(l). 设 P_0 是 F 中最短 u_0-v_0 路，P_1 是 F 中最短 u_1-v_1 路，则 P_0 与 P_1 必相交，有两种可能.

(5a) P_0 与 P_1 相交于一点 r，如图 5.5(m)，G 有同胚于 K_5 的子图；

(5b) P_0 与 P_1 相交多于一点，设从 u_0 出发沿 P_0 与 P_1 的第一个公共点是 r_1，最后一个公共点是 r_2，如图 5.5(n)，G 有子图同胚于 $K_{3,3}$.

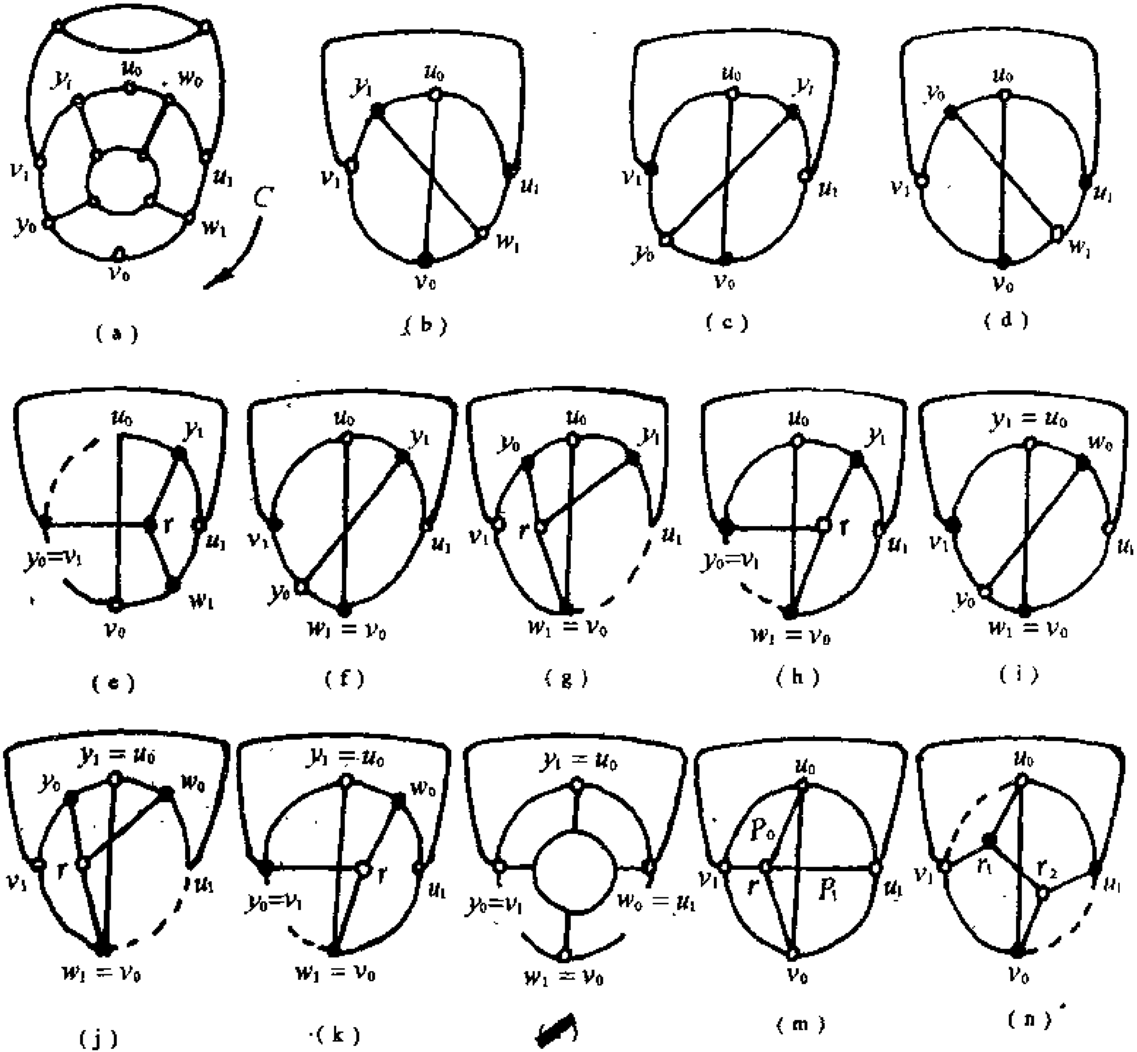


图 5.5 (a) 为一般情形，(b) 为情形 1，(c) ~ (e) 为情形 2，(f) ~ (h) 为情形 3，(i) ~ (k) 为情形 4，(l) ~ (n) 为情形 5

综上定理得证.

§ 5.4 对偶图

定义 5.6 设 G 是平面图，G 的对偶图 G^* 是如下构造的一个图：G 的面 f 对应 G^* 的顶点 f^*，G 的边 e 对应 G^* 的边 e^*，在 G^* 中有边 $e^* = (f^*, g^*)$ 当且仅当在 G 中面 f 与面 g 有公共边界 e.

例 5.8 给出平面图 G 如图 5.6(a). G 有 6 个面 f_i, $i = 1, \cdots, 6$，相应地 G^* 有 6 个顶点

f_i^*, $i=1,\cdots,6$. 顶点 f_i^* 与 f_j^* 相邻，当且仅当面 f_i 与面 f_j 相邻，由此得对偶图 G^* 如图 5.6(b).

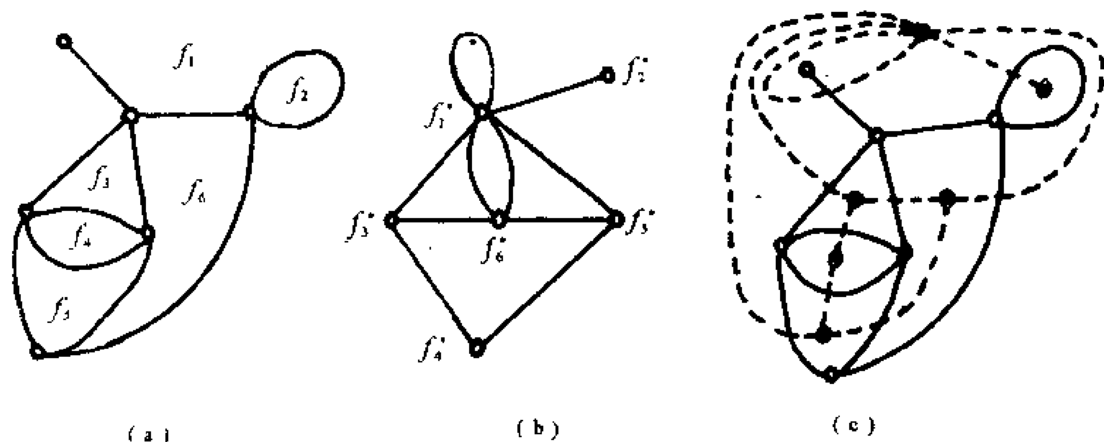


图 5.6 (a) 平面图 G (b) 对偶图 G^* (c) 对偶图的自然平面嵌入

从定义 5.6 直接得到：

$$|V(G^*)| = |F(G)|, \tag{5.12}$$

$$|E(G^*)| = |E(G)|, \tag{5.13}$$

$$d_{G^*}(f^*) = d_G(f),\ f \in F(G). \tag{5.14}$$

定理 5.15 平面图 G 的对偶图 G^* 是连通平面图.

证 平面图 G 的任意两个面 f, g，都有面 $f_1, f_2, \cdots, f_k (k \geqslant 2)$ 使 $f_1 = f, f_k = g$，且 f_i 与 f_{i+1} 相邻，$i = 1, \cdots, k-1$，故 G^* 的任意两个顶点是连通的，所以 G^* 是连通的.

关于 G^* 的平面性，可以从 G^* 的一个自然平面嵌入法得出，如图 5.6(c) 所示，即

(1) 在 G 的每个面 f 内放置顶点 f^*；

(2) 若 G 的面 f 与 g 有公共边 e，则对应于 G^* 的顶点 f^* 与 g^* 间连接一条线 e^*，使 e^* 恰好穿过边 e 一次而不与 G 的其它边相交.

定理得证.

由于一个可平面图可能不只一种平面嵌入，故平面图的对偶图不一定唯一(习题 5.24). 但我们有下面的定理.

定理 5.16 设 G 是平面图，则 G^{**} 与 G 同构的充要条件是 G 是连通的.

证 因为 G^{**} 是连通的，故必要性得证. 今证充分性. 设 G 是连通平面图，由 Euler 公式，

$$|V(G)| - |E(G)| + |F(G)| = |V(G^*)| - |E(G^*)| + |F(G^*)|,$$

代入式(5.12) 和(5.13) 得

$$|F(G^*)| = |V(G)|. \tag{5.15}$$

设 e^* 是 G^* 中一个面的边界，由 G^* 的自然平面嵌入法，在 G 中有边 e 与 e^* 相交且不与 G^* 中其它边相交，e 的端点必有一个在 G^* 的该面中，再由式(5.15)，G^* 的每个面恰有 G 的一个顶点. 按照 G^{**} 的自然平面嵌入法，G^* 的每个面也恰有 G^{**} 的一个顶点. 因此，可把 G^*

中各面内的 G^{**} 的顶点就取为 G 的顶点，同时，G^{**} 的边 e^{**} 也可取为 G 中连接相同两个端点的边 e. 在这种情形，G^{**} 就是 G，换言之，G^{**} 与 G 同构. **证毕.**

定理 5.16 指出，在同构的意义下，连通平面图的对偶图的对偶图就是自身. 虽然连通平面图的对偶不一定唯一(参见习题 5.24)，即两个不同构的图可能其对偶图是同构的，但对偶的对偶是唯一的，即连通平面图和它的对偶图是互为对偶的.

从图 5.6(c) 我们也可看到，若把虚线的图作为给定的图，则实线的图就是对偶图的平面嵌入. 换言之，可以把 G 看作是 G^{**} 对 G^* 的一个自然平面嵌入.

§5.5 平面图的其它刻划

定义 5.7 称图 G^* 是图 G 的组合对偶，如果

(1) $E(G)$ 与 $E(G^*)$ 有一个一一对应，

(2) 对任意边集 $S \subseteq E(G)$ 和 $S^* \subseteq E(G^*)$，都有

$$\xi(G-S)=\xi(G)-\eta(G^*[S^*]). \tag{5.16}$$

定理 5.17 设 G^* 是 G 的组合对偶，则

$$\xi(G)=\eta(G^*), \tag{5.17}$$

$$\eta(G)=\xi(G^*). \tag{5.18}$$

证 令 $S=E(G)$，则由式(5.16) 推出式(5.17). 由定理 3.1，有

$$\xi(G^*)+\eta(G^*)=|E(G^*)|=|E(G)|=\xi(G)+\eta(G),$$

再代入式(5.17) 即得式(5.18). **证毕.**

定理 5.18 设 G^* 是 G 的组合对偶，则 G 是 G^* 的组合对偶，即 G 与 G^* 互为组合对偶.

证 需证

$$\xi(G^*-S^*)=\xi(G^*)-\eta(G[S]),$$

即需证

$$\xi(G^*[S^*])=\xi(G^*)-\eta(G-S). \tag{5.19}$$

事实上，

$$\begin{aligned}\xi(G^*[S^*]) &= |E(G^*[S^*])|-\eta(G^*[S^*])\\ &= |E(G^*[S^*])|-\xi(G)+\xi(G-S)\\ &= |E(G^*[S^*])|-\eta(G^*)+|E(G-S)|-\eta(G-S)\\ &= |E(G^*)|-\eta(G^*)-\eta(G-S)\\ &= \xi(G^*)-\eta(G-S),\end{aligned}$$

故得式(5.19). **证毕.**

定义 5.7 的组合对偶可以看作定义 5.6 的对偶的一种抽象，而后者一般称为几何对偶.

H. Whitney[50][51] 给出了可平面性的另一刻划：

定理 5.19 G 是可平面的，当且仅当存在 G 的组合对偶.

定义 5.8 删去图 G 的边 e 并使 e 的两个端点重合，称为对边 e 的缩并，记为 $G\cdot e$. 称图 G' 是 G 的缩并图，若 G' 是图 G 经有限次边的缩并得到的.

下面的可平面性刻划是 K. Wagner[52] 给出的.

定理 5.20 G 是可平面的，当且仅当 G 没有缩并图的基图是 K_5 或 $K_{3,3}$ 的子图.

例 5.9 图 5.7(a) 的图称为 Petersen 图，它是不可平面的，因为它的缩并图的基图是 K_5，如图 5.7(b).

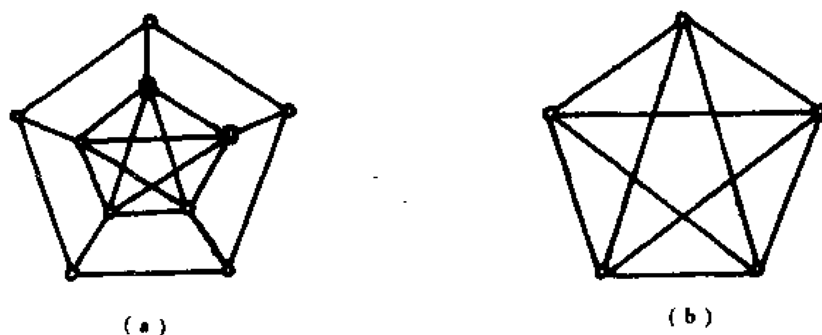


图 5.7 (a) Petersen 图 (b) 缩并图的基图

用圈的结构来刻划可平面性属于 S. MacLane[54].

定理 5.21 图 G 是可平面的，当且仅当 G 的每一个至少有三个顶点的块有一个圈基 $C_1,\cdots,C_r$ 和另外一个圈 C_0，使每条边恰好在这 $r+1$ 个圈的两个圈上.

以上讨论得到可平面性的下列等价条件：

(1) G 是可平面的；

(2) G 没有同胚于 K_5 或 $K_{3,3}$ 的子图；

(3) G 没有缩并图的基图是 K_5 或 $K_{3,3}$ 的子图；

(4) G 有一个组合对偶；

(5) G 的每个至少有三个顶点的块有一个圈基 $C_1,\cdots,C_r$ 和另一个圈 C_0，使每条边恰好在这 $r+1$ 个圈的两个圈上.

§5.6 应用——电网络方程

如果把导线和若干电子元件相连接则得到一个电网络. 每个元件有二个端点与别的元件连接，这些端点称为节点. 一个电网络的电路图可用一个有向图表示，图的顶点与边分别对应电网络的节点与元件，边的方向表示电流的方向，并规定电流从节点流出为正，流入为负.

在一个电网络中有两个基本定律：

(1) Kirchhoff 电流定律(KCL)：对任一节点，流入或流出电流的代数和为零.

(2) Kirchhoff 电压定律(KVL)：对任一回路，其电压降的代数和为零.

当用一个有向(n,m)图 G 表示电网络时，可以把 KCL 与 KVL 写成矩阵形式：

$$KCL \quad AI_m(t)=0, \tag{5.20}$$

$$KVL \quad BV_m(t)=0, \tag{5.21}$$

其中 A 是图 G 的关联矩阵，B 是图 G 的圈矩阵，

$$I_m(t)=(i_1(t),i_2(t),\cdots,i_m(t))^T$$

$$V_m(t)=(v_1(t),v_2(t),\cdots,v_m(t))^T$$

分别是 m 维电流列向量和电压列向量，t 表示时间. 因为余圈空间即是余圈矩阵的行空间，又是关联矩阵的行空间，所以式(5.20)的关联矩阵可用余圈矩阵 Q 代替，即得

$$QI_m(t)=0. \tag{5.22}$$

例 5.10 如图 5.8(a) 的电网络对应有向图 G，如图 5.8(b). 有向图 G 中顶点 v_i 对应电网络中节点 i. 有向图 G 中边的方向表示电流在电网络中的方向.

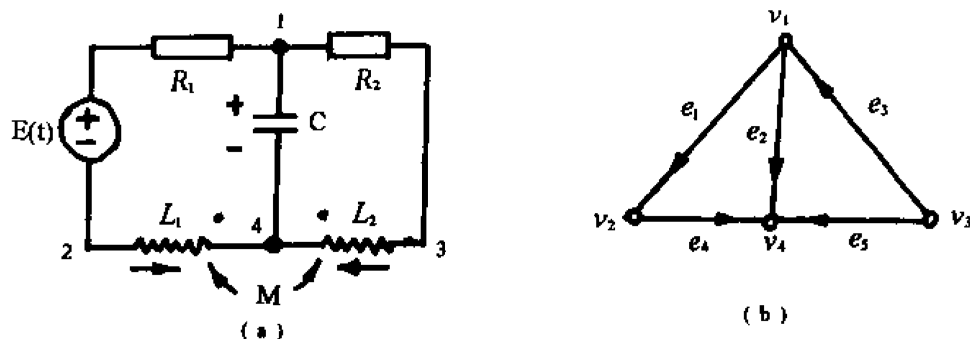


图 5.8 (a) 电网络 N (b) 对应有向图 G

根据图 G 可分别写出 KCL 与 KVL 矩阵方程如下：

KCL $\quad AI_m(t)=0$，

$$\begin{pmatrix}1&1&-1&0&0\\-1&0&0&1&0\\0&0&1&0&1\\0&-1&0&-1&-1\end{pmatrix}\begin{pmatrix}i_1(t)\\i_2(t)\\i_3(t)\\i_4(t)\\i_5(t)\end{pmatrix}=0. \tag{5.23}$$

或 $\quad QI_m(t)=0$，（树枝 e_1,e_2,e_3），

$$\begin{pmatrix}1&0&0&-1&0\\0&1&0&1&1\\0&0&1&0&1\\1&1&0&0&1\\0&1&-1&1&0\\1&1&-1&0&0\end{pmatrix}\begin{pmatrix}i_1(t)\\i_2(t)\\i_3(t)\\i_4(t)\\i_5(t)\end{pmatrix}=0. \tag{5.24}$$

KVL $\quad BV_m(t)=0$，（连枝 e_4,e_5），

$$\begin{pmatrix}1&-1&0&1&0\\0&-1&-1&0&1\\1&0&1&1&-1\end{pmatrix}\begin{pmatrix}v_1(t)\\v_2(t)\\v_3(t)\\v_4(t)\\v_5(t)\end{pmatrix}=0. \tag{5.25}$$

容易看到，方程(5.23)，(5.24)与(5.25)都不独立. 因为 $r(A)=r(Q)=n-1$，$r(B)=m-n+1$，在例 5.10，$n=4$，$m=5$，所以 KCL 只有 3 个独立方程，KVL 只有 2 个独立方程.

KCL 与 KVL 的等价独立方程是

KCL $\qquad A_f I_m(t) = 0,$ (5.26)

或 $\qquad Q_f I_m(t) = 0.$ (5.27)

KVL $\qquad B_f V_m(t) = 0,$ (5.28)

其中 A_f, Q_f 与 B_f 分别表示图 G 的基本关联矩阵，基本余圈矩阵和基本圈矩阵.

例 5.11　在例 5.10 中，等价于式(5.23)，(5.24) 与(5.25) 的独立方程是：

KCL $\qquad A_f I_m(t) = 0,$

$$\begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \end{pmatrix} = 0. \tag{5.29}$$

或 $\qquad Q_f I_m(t) = 0,$

$$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \end{pmatrix} = 0. \tag{5.30}$$

KVL $\qquad B_f V_m(t) = 0,$

$$\begin{pmatrix} 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \\ v_5(t) \end{pmatrix} = 0. \tag{5.31}$$

由式(3.33)，(3.35) 和(3.34)，可把 A_f, Q_f 与 B_f 写为分块形式：

$$A_f = (A_f|\overline{T}, A|T) = (A_{f,1}, A_{f,2}). \tag{5.32}$$

$$Q_f = (Q_f|\overline{T}, Q_f|T) = (Q_{f,1}, I_{n-1}). \tag{5.33}$$

$$B_f = (B_f|\overline{T}, B_f|T) = (I_{m-n+1}, B_{f,2}). \tag{5.34}$$

其中 T 是 G 的生成树，$\overline{T}$ 是余树.

式(5.28) 改写为

$$(I_{m-n+1}, B_{f,2}) \begin{pmatrix} V_{m,1}(t) \\ V_{m,2}(t) \end{pmatrix} = 0,$$

即

$$V_{m,1}(t) = -B_{f,2} V_{m,2}(t), \tag{5.35}$$

其中 $V_{m,1}(t), V_{m,2}(t)$ 分别表示连枝电压，树枝电压.

由式(3.36)，$Q_{f,1} = -B_{f,2}{}^T$，代入式(5.35) 得

$$V_{m,1}(t) = Q_{f,1}{}^T V_{m,2}(t).$$

于是

$$\begin{pmatrix} V_{m,1}(t) \\ V_{m,2}(t) \end{pmatrix} = \begin{pmatrix} Q_{f,1}{}^T \\ I_{n-1} \end{pmatrix} V_{m,2}(t) = Q_f^T V_{m,2}(t),$$

即

$$V_m(t) = Q_f^T V_{m,2}(t). \tag{5.36}$$

式(5.36)称为**余圈转换矩阵方程**,它表明电网络中支路电压可以由基本余圈的树枝电压线性表出.

式(5.27)改写为

$$(Q_{f,1}, I_{n-1}) \begin{pmatrix} I_{m,1}(t) \\ I_{m,2}(t) \end{pmatrix} = 0,$$

即

$$I_{m,2}(t) = -Q_{f,1} I_{m,1}(t). \tag{5.37}$$

其中 $I_{m,1}(t)$, $I_{m,2}(t)$ 分别表示连枝电流,树枝电流. 由式(3.36), $Q_{f,1} = -B_{f,2}{}^T$,代入式(5.37)得

$$I_{m,2}(t) = B_{f,2}{}^T I_{m,1}(t). \tag{5.38}$$

于是

$$\begin{pmatrix} I_{m,1}(t) \\ I_{m,2}(t) \end{pmatrix} = \begin{pmatrix} I_{m-n+1} \\ B_{f,2}{}^T \end{pmatrix} I_{m,1}(t) = B_f{}^T I_{m,1}(t),$$

即

$$I_m(t) = B_f^T I_{m,1}(t). \tag{5.39}$$

式(5.39)称为**圈转换矩阵方程**,它表明电网络中支路电流可由基本圈的连枝电流线性表出.

把式(5.35)与(5.38)合并在一起写出,有

$$\begin{pmatrix} V_{m,1}(t) \\ I_{m,2}(t) \end{pmatrix} = \begin{pmatrix} 0 & -B_{f,2} \\ B_{f,2}{}^T & 0 \end{pmatrix} \begin{pmatrix} I_{m,1}(t) \\ V_{m,2}(t) \end{pmatrix}. \tag{5.40}$$

式(5.40)说明,m 条边的电流和电压,只要知道其中连枝电流和树枝电压便可求出其它.

例 5.12 在例 5.10 中我们选取图 5.8(b) 的边 e_1, e_2, e_3 为树枝,e_4, e_5 为连枝,得

$$B_{f,2} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & -1 & -1 \end{pmatrix}.$$

于是方程(5.40)对图 5.8(a) 的电网络有

$$\begin{pmatrix} v_4(t) \\ v_5(t) \\ i_1(t) \\ i_2(t) \\ i_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_4(t) \\ i_5(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{pmatrix}. \tag{5.41}$$

容易看到,式(5.41)就是合并式(5.31)与(5.30)的结果.

由式(5.28), $V_m(t)$ 正交于 B_f 的行空间,即正交于圈空间,由定理 3.11, $V_m(t)$ 是势差.于是,由式(3.18),势差 $V_m(t)$ 可写为

$$V_m(t) = A_f^T V_n(t), \tag{5.42}$$

其中A_f是基本关联矩阵，$V_t(t)$是相应$(n-1)$个节点的节点电压向量. 式(5.42)称为**节点转换矩阵方程**.

当电网络是平面网络时，相应的有向图是平面图，由定理 5.3，有向图的所有内部面的周线构成圈基. 我们称平面图的内部面为**网孔**，称内部面的周线构成的矩阵为**网孔矩阵**，记为B_η.

在式(5.28)中用B_η代替B_f，得 KVL 的网孔矩阵方程：

KVL
$$B_\eta V_m(t)=0. \tag{5.43}$$

在式(5.39)中用B_η代替B_f，得圈转换矩阵方程的网孔矩阵形式：

$$I_m(t)=B_\eta^T I_\eta(t), \tag{5.44}$$

这里$I_\eta(t)$是网孔电流. 称式(5.44)为**网孔矩阵转换方程**，它表明支路电流可由网孔电流线性表出.

例 5.13　图 5.9 是相应于平面电网络的有向图. 试写出余圈转换矩阵方程，圈转换矩阵方程，节点转换矩阵方程，KVL 的网孔矩阵方程和网孔矩阵转换方程.

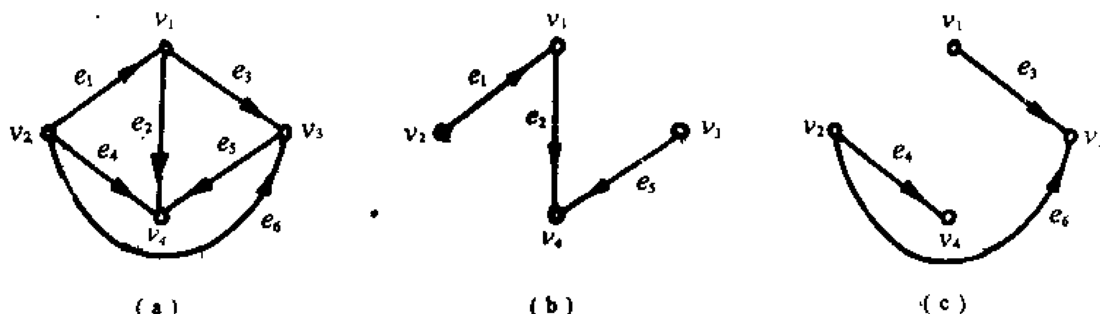


图 5.9　(a) 平面网络图　(b) 生成树　(c) 余树

首先写出基本关联矩阵，基本余圈矩阵，基本圈矩阵和网孔矩阵：

$$A_f=\begin{pmatrix}-1&1&1&0&0&0\\1&0&0&1&0&1\\0&0&-1&0&1&-1\end{pmatrix}\text{（从 }A\text{ 删去 }v_4\text{ 的行），}$$

$$Q_f=\begin{pmatrix}1&0&0&1&0&1\\0&1&1&1&0&1\\0&0&-1&0&1&-1\end{pmatrix}\text{（按式(5.33)），}$$

$$B_f=\begin{pmatrix}0&-1&1&0&1&0\\-1&-1&0&1&0&0\\-1&-1&0&0&1&1\end{pmatrix}\text{（按式(5.34)），}$$

$$B_\eta=\begin{pmatrix}1&1&0&-1&0&0\\0&-1&1&0&1&0\\0&0&0&1&-1&1\end{pmatrix}\text{（网孔按顺时针方向）.}$$

然后分别写出所求方程.

余圈转换矩阵方程：$V_m(t)=Q_f^T V_{m,2}(t)$，

$$
\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \\ v_5(t) \\ v_6(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_5(t) \end{bmatrix}.
$$

圈转换矩阵方程：$I_m(t) = B_f^T I_{m,l}(t)$，

$$
\begin{bmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \\ i_6(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 \\ -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_3(t) \\ i_4(t) \\ i_6(t) \end{bmatrix}.
$$

节点转换矩阵方程：$V_m(t) = A_f^T V_n(t)$，

$$
\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \\ v_5(t) \\ v_6(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} V_{n1}(t) \\ V_{n2}(t) \\ V_{n3}(t) \end{bmatrix}.
$$

KVL 的网孔矩阵方程：$B_\eta V_m(t) = 0$，

$$
\begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \\ v_5(t) \\ v_6(t) \end{bmatrix} = 0.
$$

网孔转换矩阵方程：$I_m(t) = B_\eta^T I_\eta(t)$，

$$
\begin{bmatrix} i_1(t) \\ i_2(t) \\ i_3(t) \\ i_4(t) \\ i_5(t) \\ i_6(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} I_{\eta1}(t) \\ I_{\eta2}(t) \\ I_{\eta3}(t) \end{bmatrix}.
$$

§5.7 应用——平面性判定

在 §5.5 中给出了图的可平面性的很好的刻划，但应用它们来实际判定一个图是不是可平面的却并不容易. W. R. Dunn 和 S. P. Chan 提出了利用圈矩阵判定一个图的平面性

的算法，其优点是简单，易于在计算机上实现. 这个算法的依据是下面的定理[55].

定理 5.22 设(n,m)图G是一个块，则G可平面的充要条件是G的圈矩阵含有一个$(m-n+2)$行构成的子矩阵，其每列恰有两个"1".

如果存在圈矩阵B的一个满足定理5.22条件的子矩阵B_a，则B_a的行数正好是平面图的面数，而B_a的每一行对应于图的一个面. 直观地说，对图G所有的圈，属于网孔的圈比其它圈的边数要少些. 因此先从边数少的圈考查可以尽快找到所求的子矩阵B_a. 为此引入下面的定义.

定义 5.9 设G是(n,m)图，有e_1个圈，G的圈矩阵B的行已按每个圈所含边的数目从小到大依次排列，称这样的圈矩阵是**边递增的**. 设J是e_1维列向量，若J的i分量等于B的i行之和即i行所含"1"的个数，则称J是**边集向量**. 设M_k是e_1维行向量，若M_k有$(m-n+2)$个"1"而其余分量为零，则称M_k是**圈组合向量**. 圈组合向量的意义是，当i分量为1，表示选择B的i行，当i分量为0，表示不选择B的i行.

下面都假定圈矩阵B是递增的.

引理 2.23 设(n,m)图G是一个块，J是边集向量，若G可平面，则存在圈组合向量M_k使

$$M_kJ = 2m. \tag{5.45}$$

证 由定理5.22，有子矩阵B_a每列恰有两个"1". 设B_a对应的圈组合向量是M_k，而M_kJ正好是B_a的$(m-n+2)$个圈的边数之和，故式(5.45)成立. **证毕.**

前面已经指出，在考查圈矩阵B时从边数少的圈开始，当B的行是边递增时，这相当于系统地产生每一个圈组合向量，即依次比较两个圈组合向量的各分量，第一次出现对应分量不相等时，分量为"1"的排前，为"0"的排后，即按字典序排列. 换句话说，检验圈矩阵B就简化为产生M_k，当满足必要条件(5.45)时，再检验由M_k所选择的子矩阵是否每列恰有两个"1"，这就大大减少了工作量. 但是，要判断一个圈是不可平面的，还需引入本原圈组合向量的定义.

定义 5.10 相邻两个"1"之间所夹"0"的个数不超过一个时，这样的圈组合向量称为**本原的**.

例 5.14 图5.9(a)的基础图是一个无向图，其圈矩阵B按边递增排列是

$$B = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix},$$

边集向量是

$$J = (3,3,3,3,4,4,4,)^T.$$

因为$m=6$，$n=4$，所以$m-n+2=4$，故圈组合向量M_k有7个分量，其中有4个"1"，3个"0"，如(1111000)，(1110100)，(1001011)，(0001111)，(0100111)等，其中第1，2，4个

是本原的.

引理 5.24 给定图 G 的边集向量 J，本原组合向量 M_s，则对所有按系统生成法排在 M_s 之后的圈组合向量 $M_t(t>s)$ 都满足

$$M_tJ \geqslant M_sJ. \tag{5.46}$$

证 由系统生成法可知，在一个本原圈组合向量 M_s 之后的每一个圈组合向量 M_t，M_t 中元素"1"的位置或保持原位或更靠右，再由圈矩阵的边递增性即得式(5.46). **证毕.**

例 5.15 下面列出由系统生成法生成的所有圈组合向量：

$$
\begin{aligned}
{}^*M_1 &= (11100),\\
{}^*M_2 &= (11010),\\
M_3 &= (11001),\\
{}^*M_4 &= (10110),\\
{}^*M_5 &= (10101),\\
M_6 &= (10011),\\
{}^*M_7 &= (01110),\\
{}^*M_8 &= (01101),\\
{}^*M_9 &= (01011),\\
{}^*M_{10} &= (00111).
\end{aligned}
$$

打"*"者为本原的.在每一个本原圈组合向量之后的每个圈组合向量，元素"1"的位置或保持原位或更靠右.

引理 5.25 设(n,m)图 G 是块，边集向量是 J，对于由系统生成法产生的一列圈组合向量 $M_1,M_2,\cdots,M_s$，其中 M_s 是本原的，则当

(1) $M_1,\cdots,M_{s-1}$ 都不满足定理 5.22，

(2) $M_sJ>2m$

时，G 是不可平面的.

证 由定理 5.22 和引理 5.24 直接得证.

以上分析即得平面性判定算法.

算法 5.1 Dunn-Chan 平面性判定的算法

设(n,m)图 G 是块，

第 1 步 生成边递增圈矩阵 B.

第 2 步 生成边集向量 J.

第 3 步 第一个圈组合向量是

$$M_1=(\underbrace{11\cdots1}_{m-n+2}0\cdots0),$$

令 $k=1$.

第 4 步 计算 M_kJ. 若小于 $2m$，转第 5 步；若等于 $2m$，转第 6 步；若大于 $2m$，转第 7 步.

第 5 步　令 $k = k + 1$，生成 M_k，转第 4 步.

第 6 步　设由 M_k 所选择的 B 的 $m - n + 2$ 行构成的子矩阵是 B_a，若 B_a 的每列恰有两个“1”，则 G 是可平面的，停. 否则转第 5 步.

第 7 步　若 M_k 是本原的，则 G 不可平面，停. 否则转第 5 步.

关于平面性的有效算法参看文献[39] ~ [42].

习 题 五

5.1　不利用定理 5.3 而直接证明 Euler 公式(5.2).

5.2　(1) 试证在简单图中，任一顶点的度不大于其余顶点的度之和；

(2) 举例说明在平面图中，可能有某个面的度大于其余面的度之和.

5.3　证明定理 5.10.

5.4　简单平面图 G，存在 $v \in V(G)$，$d(v) \leqslant 5$.

5.5　设 G 是 2- 连通简单平面图且没有三角形，则

$$|E(G)| \leqslant 2|V(G)| - 4.$$

5.6　G 是简单平面图，每个面的边界是一个 k 圈，则

$$|E(G)| = \frac{k}{k-2}(V(G) - 2).$$

5.7　证明定理 5.11.

5.8　画出所有 6 个顶点的极大可平面图.

5.9　设 G 是简单平面图，$|E(G)| < 30$，则存在顶点 v，$d(v) \leqslant 4$.

5.10　设 G 是简单连通平面图，$|V(G)| = 7$，$|E(G)| = 15$，则 $d(f) = 3$.

5.11　设 G 是可平面图，证明 G 的平面嵌入有相同的面数，并举例说明面的度未必相同.

5.12　任意图 G 均能嵌入 R^3.

5.13　图 G 可以嵌入平面的充要条件是 G 能嵌入球面.

5.14　不可平面图 $K_{3,3}$ 可以嵌入环面.

5.15　不可平面图 K_5 可以嵌入环面.

5.16　设 G 是简单平面图，$|V(G)| \geqslant 4$，则 G 至少有 4 个顶点的度不超过 5.

5.17　设 G 外可平面，$|V(G)| \geqslant 2$，且没有三角形，则

$$|E(G)| \leqslant (3|V(G)| - 4)/2.$$

5.18　设 G 是极大外平面图，$|V(G)| \geqslant 3$，顶点都在外部面上，则 G 有 $(|V(G)| - 2)$ 个内部面.

5.19　设 G 是极大外平面图，$|V(G)| \geqslant 3$，则 G 至少有 3 个顶点的度不超过 3，至少有 2 个顶点的度等于 2.

5.20　K_4 和 $K_{2,3}$ 都不是外可平面的.

5.21　画出所有 7 个顶点的极大外平面图.

5.22　每一个至少有 7 个顶点的外可平面图的补图不是外可平面的，而 7 是这种数目中最小的.

5.23　证明在平面图 G 及对偶图 G^* 中，G 的桥对应 G^* 的环，G 的环对应 G^* 的桥.

5.24　举例说明平面图的对偶图不是唯一的.

5.25　一个平面图若与自己的对偶图同构，则称该图为自对偶图. 设 G 是自对偶图，则

$$|E(G)| = 2|V(G)| - 2.$$

5.26　分别作出简单自对偶图和不是简单图的自对偶图.

5.27　设 G 是不可平面图而任删去一边都成为可平面的，则称 G 为极小不可平面图. 试证极小不可平面图是一个简单块.

5.28　设 G 是简单平面图，$|V(G)| \geq 3$，则下列四条等价

(1) G 是极大平面图；

(2) G 的每个面是三角形；

(3) $|F(G)| = 2|V(G)| - 4$；

(4) $|E(G)| = 3|V(G)| - 6$.

5.29　(1) 设 G 是简单平面图，$|V(G)| \geq 11$，则 G^c 不可平面；

(2) 举例说明 $|V(G)| = 8$，G 和 G^c 都可平面.

5.30　设 G 是平面图，G^* 是对偶图，则

$C = [e_1, \cdots, e_k]$ 是 G 的余圈 $\Longleftrightarrow C^* = [e_1^*, \cdots, e_k^*]$ 是 G^* 的圈.

5.31　设 G 是平面图，G^* 是对偶图，则

$C = [e_1, \cdots, e_k]$ 是 G 的圈 $\Longleftrightarrow C^* = [e_1, \cdots, e_k]$ 是 G^* 的余圈.

5.32　设 G 是平面图，G^* 是对偶图，则

$$\xi(G) = \eta(G^*),\ \eta(G) = \xi(G^*).$$

5.33　设 G 是简单连通平面图，G^* 是对偶图，则

$T = [e_1, \cdots, e_k]$ 是 G 的生成树 $\Longleftrightarrow T^* = [e_1^*, \cdots, e_k^*]$ 是 G^* 的余树.

5.34　设 G 是简单连通平面图，G^* 是对偶图，则

$T = [e_1, \cdots, e_k]$ 是 G 的余树 $\Longleftrightarrow T^* = [e_1^*, \cdots, e_k^*]$ 是 G^* 的生成树.

5.35　(1) 5 个面的平面图至少有 2 个面不相邻；

(2) 画出 4 个面的平面图使任二面相邻.

5.36　设 $S = \{x_1, \cdots, x_n\}$，$n \geq 3$，是平面上一个点集，且任意二点间的距离大于等于 1，证明最多有 $(3n - 6)$ 个点对，它们之间的距离为 1.

5.37　不存在 6- 连通简单平面图.

5.38　5- 连通简单平面图至少有多少个顶点？试构造一个 5- 连通的简单平面图.

5.39　如图 5.10，证明 Heawood 图是不可平面的.

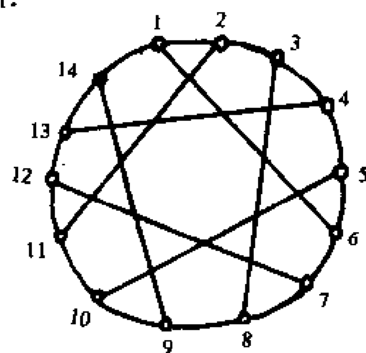


图 5.10　Heawood 图

5.40　给出图 G 的基本关联矩阵

$$A_f = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix},$$

试画出图 G 和对偶图 G^*.

5.41　给出图 G 的圈矩阵

$$B=\begin{bmatrix}1&1&0&1&0&1\\0&0&0&0&1&1\\0&0&1&1&0&1\\1&1&1&0&0&0\\0&0&1&1&1&0\\1&1&0&1&1&0\end{bmatrix},$$

试画出图 G 和对偶图 G^*.

5.42　以 $\tau(G)$ 表连通图 G 的生成树数目，则

$$\tau(G)=\tau(G-e)+\tau(G\cdot e).$$

5.43　利用习题 5.42 的公式计算 $\tau(K_4)$.

5.44　利用习题 5.42 的公式证明 Cayley 公式：

$$\tau(W_n)-4\tau(W_{n-1})+4\tau(W_{n-2})-\tau(W_{n-3})=0,\ (n\geqslant 6),$$

其中 W_n 是 $(n-1)$ 圈内一顶点向该圈所有 $(n-1)$ 个顶点各引一条边所构成的图，称为 n-轮.

5.45　图 5.11 是平面电网络相应的有向图，试写出余圈转换矩阵方程，圈转换矩阵方程，节点转换矩阵方程，KVL 的网孔矩阵方程和网孔转换矩阵方程，这里生成树为 $T=[e_1,e_2,e_4,e_5]$，网孔取顺时针方向.

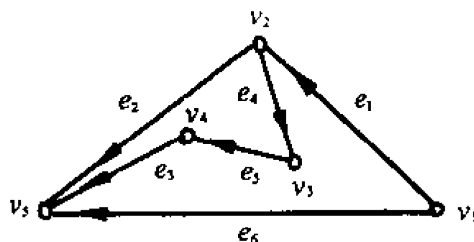


图 5.11

5.46　用 Dunn-Chan 平面性判定算法证明 K_5 是不可平面的，而 K_5 任去一边是可平面的.

5.47　用 Dunn-Chan 平面性判定算法证明 $K_{3,3}$ 是不可平面的，而 $K_{3,3}$ 任去一边是可平面的.

第六章　图的基本算法

§6.1　图的算法与有效性

在 §1.1 中曾经提到环球旅行问题，即在一个完全图中找一个含有所有顶点的圈．这样的圈称为哈密顿圈．

定义 6.1　包含图 G 的所有顶点的路（圈）称为**哈密顿路（圈）**．含有哈密顿圈的图称为**哈密顿图**．

例 6.1　假设有 n 个城市，已知任意两个城市间的旅行费用．今有旅行推销员从某城市出发，欲到其余 $(n-1)$ 个城市去推销．问应选择怎样的路线，使其余 $(n-1)$ 个城市刚好各访问一次而又回到出发城市，其总费用最少？这个问题被称为**旅行推销员问题**．

若以每个城市为顶点，任意两个城市间的费用作为连接相应两个顶点的边的权，当某两个城市不能直通时，相应边的权设为任意大，由此得到 n 个顶点的赋权完全图．例 6.1 的旅行推销员问题便转化为在赋权完全图中求权最小的哈密顿圈．

注意到在 K_n 中有 $\frac{1}{2}(n-1)!$ 个哈密顿圈，若用穷举法求最小权哈密顿圈，则需计算 $\frac{1}{2}(n-1)!$ 个哈密顿圈的权并从中求出权最小的．假定以二十个城市计，即 $n=20$，如果计算机每秒能处理 10^8 个圈，则处理完 20 个顶点的完全图，总共需要时间为

$$\frac{\frac{1}{2}\cdot 19!}{3.1536\times 10^7\times 10^8}\approx 19(\text{年}),$$

显然这是无法得出结果的．换言之，穷举法只能在规模比较小的某些情形才是可能的．因此，对图论的问题必须寻求合适的有效的算法．

图的算法是在计算机上进行的．一个算法就是一组有穷的规则，根据这些规则，准确地告诉我们在每一步该做什么，怎样开始，怎样结束．一个算法的表述可以有三种方式：

(1) 用运行步骤表述；

(2) 用运行框图表述；

(3) 用算法语言表述；

本书采用运行步骤表述．一个算法应当包括五个重要方面：

(1) 有限性，即算法必须在有限步内终止，

(2) 确定性，即每个步骤都是唯一确定而没有歧意的，

(3) 算法开始时的输入，

(4) 算法结束时的输出，

(5) 算法的有效性．

正如本节开始时所述，在图论算法中，有效性是极为重要的．算法的有效性决定于问题的存贮量和计算时间两个因素．这两个因素都可作为输入规模的函数．对于图论问题，输入规模是顶点数 $|V(G)|$ 和边数 $|E(G)|$．一般说，影响图论算法有效性的不是存贮量而是计算时间．在估计某个算法的优劣时，可以比较最坏情形的执行时间，最好情形的执行时间，或平均情形的执行时间．把算法需要的执行时间表成输入规模的函数，称为**时间复杂性**．假定问题的规模用一个正整数 n 来表示，若执行时间 $t \leqslant an^k$，其中 a,k 为常数，则时间复杂性是 $O(n^k)$，并称该算法是**多项式时间复杂性算法**．时间复杂性为 $O(a^n)$ 的算法，其中 a 为常数称为**指数时间复杂性算法**．多项式时间复杂性算法称为**有效算法或好的算法**，而指数时间复杂性算法是**非有效算法或不好的算法**．对于规模较小的问题，指数时间复杂性算法还是可用的．但随着规模的增加，指数时间复杂性算法就完全无用了，如表 6.1.

表 6.1　时间复杂性比较表（假设计算机每秒 10^6 次）

复杂性 \ 规模	$n=10$	$n=20$	$n=30$	$n=50$	$n=100$
n^2	0.000001 秒	0.000004 秒	0.000009 秒	0.000025 秒	0.0001 秒
2^n	0.00001 秒	0.01 秒	10.74 秒	130 天	4×10^{14} 年
$n!$	0.036288 秒	771 年	8.4×10^{16} 年	9.6×10^{48} 年	3.1×10^{142} 年

关于时间复杂性问题还可以从另一角度加以比较，即当计算机的运算速度提高时，计算机每小时处理的规模的增加情形，如表 6.2.

表 6.2　每小时处理规模比较表

复杂性 \ 速度	10^6 次 / 秒	10^8 次 / 秒	10^9 次 / 秒
n^2	$n=6\times10^4$	$n=6\times10^5$	$n=1.89\times10^6$
2^n	$n=31$	$n=38$	$n=41$
$n!$	$n=12$	$n=14$	$n=15$

从表 6.2 可知，复杂性为 n^2 的多项式算法，当计算机速度提高 10 倍，每小时处理规模可以增加 3 倍以上．而对指数复杂性算法（表中 2^n，$n!$），处理规模只能增加一个常数．很明显，多项式算法更能利用计算机速度提高所带来的好处，而指数算法却不能分享到什么．

多项式算法还有一个优点，即有限个多项式算法结合而成的算法仍然是多项式算法．

对图论的算法一般分为三类．

第 1 类　多项式算法问题，即存在多项式时间复杂性算法的问题．如连通性算法，强连通性算法，生成树算法，基本圈算法，可分性算法，平面性算法，最短路算法，最小生成树算法等．

第 2 类　指数算法问题，即问题本身没有多项式算法而必须在指数时间内解决的问题，如求全部生成树，求所有圈，求全部完全子图等．

第 3 类　至今只有指数时间算法而没有找到多项式时间算法的问题．第 3 类与第 2 类不同，第 2 类是问题本身就是指数复杂性的．第 3 类不是问题本身所固有的性质，即并不能证明这些算法没有多项式时间算法，而是还没有找到．这一类的问题如：

(1) k- 团问题：在图中找基数为 k 的团（k- 团的定义见习题 4.33）．

(2) k- 独立集问题:在图中找基数为 k 的独立集(独立集的定义见习题 4.26).

(3) k- 覆盖问题:在图中找基数不超过 k 的覆盖(覆盖的定义见定义 4.3).

(4) 求图的最小控制集. 图 G 的顶点子集 S 称为**控制集**,如果每个顶点 v,或 $v \in S$,或 v 与 S 中顶点相邻. 顶点数最少的控制集称为**最小控制集**.

(5) 求图的色数. 若能用 k 种颜色给图的所有顶点着色使任意相邻两顶点有不同的颜色,则称该图是 **k- 可着色的**. 图 G 的**色数**就是 k- 可着色的数 k 的最小值,用 $\chi(G)$ 或 χ 表示 G 的色数.

(6) 求图的哈密顿圈.

(7) 求有向图的有向哈密顿圈.

(8) 旅行推销员问题.

(9) Steiner 树问题,即在一个赋权连通图中已知顶点子集 S,找一个连通 S 的所有顶点的权最小的树.

(10) 给定一个有向图 G,找出一个边的最小集合 $S \subseteq E(G)$,称为**最小反馈边集**,使 $G - S$ 没有有向圈.

(11) 给定一个有向图 G,找出一个顶点的最小集 $S \subseteq V(G)$,称为**最小反馈顶点集**,使 $G - S$ 没有有向圈.

(12) 图的同构问题.

本章讨论图的基本算法. 对于算法复杂性仅作简单讨论. 更为详细的讨论可参看文献[12],[34],[56] ~ [60].

§ 6.2 图在计算机中的表示

图在计算机中的表示通常有下列几种方法,选用哪一种更合适,与所要解决的问题及算法有关.

方法 1 邻接矩阵

设 G 是 (n,m) 图,则用一个 $n \times n$ 数组 $H(I,J)$ 存贮 G 的邻接矩阵,其 i 行 j 列的元即 (i,j) 元为 1 表示顶点 j 是顶点 i 的后继,否则 (i,j) 元为 0. 这里邻接矩阵只表示顶点的邻接关系,因此不能表示平行边,但能表示环.

对无向图,邻接矩阵是对称的,故只需存贮一个上三角阵. 一般说,存贮量的节省是以增加程序的复杂性和计算量为代价的.

对赋权图,数组 $H(I,J)$ 的 (i,j) 元之值为边 (i,j) 的权.

方法 2 关联矩阵

设 G 是 (n,m) 图,则用一个 $n \times m$ 数组 $A(I,J)$ 存贮 G 的关联矩阵,其 (i,j) 元为 1 表示边 j 是顶点 i 的出边,为 -1 表示边 j 是顶点 i 的入边,否则为 0. 对无向图,(i,j) 元为 1 表示边 j 与顶点 i 关联,否则为 0. 由于关联矩阵表示的是边与顶点的关联关系,故关联矩阵不能表示环,但能表示平行边.

关联矩阵每列有 2 个非零元,其余都是零元,因此一般这是一个稀疏矩阵,可采取比较紧凑的数据结构.

通常,一个图的边数大于顶点数,故关联矩阵的存贮量是大于邻接矩阵的.

方法 3　边目录法

直接用顶点对列出图的所有的边，对无向图，顶点对是无序的；对有向图，顶点对是有序的.

边目录法可以表示平行边，也可以表示环. 对于稀疏矩阵，这种方法较节省存贮空间，但对于图在计算机中的修改与运算可能是不方便的.

方法 4　二数组法

用两个 $|E(G)|$ 维数组，$D_1(I)$ 与 $D_2(I)$，对无向图，边 e_i 的两个端点分别在 $D_1(I)$ 与 $D_2(I)$ 的第 i 个位置，对有向图，边 e_i 的尾在 $D_1(I)$ 的第 i 个位置，头在 $D_2(I)$ 的第 i 个位置.

对赋权图 G，则再增加一个 $|E(G)|$ 维数组 $D_3(I)$，边 e_i 的权在 $D_3(I)$ 的第 i 个位置.

这种方法表达的信息量多，占存贮空间小，使用方便，因此用得较广泛.

方法 5　后继表法

设 G 是 (n,m) 图，则用 n 个一维数组列出每个顶点的后继表（对无向图即邻域表）. 第 i 个一维数组的第一个元放置顶点 i，其余的元放置顶点 i 的后继，$i = 1,2,\cdots,n$.

对于无向图，由于列出每个顶点的邻域，因此每条边都出现两次.

例 6.2　对第一章图 1.8(a) 的无向图，在计算机中的三种存贮法如下：

边目录法

$$(1,2),(1,3),(1,3),(2,2),(2,4),(3,4),(4,5),(3,5),(5,6)$$

二数组法

$$D_1(I) = (1,1,1,2,2,3,4,3,5)$$
$$D_2(I) = (2,3,3,2,4,4,5,5,6)$$

后继表法

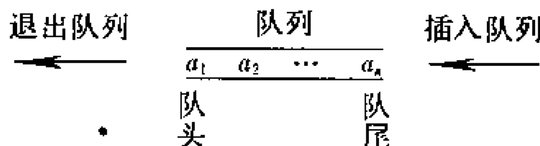
1：2,3,3　　2：1,2,4
3：1,1,4,5　　4：2,3,5
5：3,4,6　　6：5

§ 6.3　图的遍历

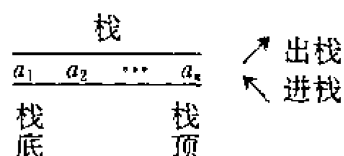
图的**遍历**又称图的**行遍**，即从图的某一顶点出发，沿图的边访问其它顶点，使图的每个顶点恰好被访问一次的过程. 例如在习题 2.59 中对二元有序正则树的三种行遍法，在第四章关于匹配的算法中的树生长过程，这些都是图的遍历问题.

图的遍历一般有两种方法：广度优先搜索与深度优先搜索.

定义 6.2　**队列**是一个先进先出的线性表. 它只允许在线性表的一端插入元素，在另一端退出元素. 允许插入的一端叫**队尾**，允许退出的一端称为**队头**，如下所示：



定义 6.3　**栈**是一个后进先出的线性表. 它只允许在线性表的同一端插入或退出，该端称为**栈顶**，而另一端称为**栈底**，如下所示：



从定义可知，队列相当于排队等候服务，谁先到谁先服务，后到的人排在队尾. 当排在队头的已经服务后即离开队列，排在后面的再依次递进等待服务. 栈相当于铁路调度站，列车依次进站等候，最先进站的列车进入栈底等候，最后进站的列车在栈顶等候. 出站时，一定是排在栈顶的列车先离站，最后离开调度站的必是最先进站的列车.

算法 6.1 遍历图的广度优先搜索法

假设从访问顶点 v_1 开始，然后访问 v_1 的未被访问的所有后继，再依次访问这些后继顶点的未被访问的所有后继，直到图的已被访问的顶点的后继都被访问到. 此时，若图的顶点都被访问，则搜索终止，否则选择一个未被访问的顶点，重复上述过程，直到图中所有顶点都被访问时迭代终止. 换句话说，广度优先搜索遍历图的过程是以 v_1 为起点，由近及远，依次访问与 v_1 有长为 1,2,… 的顶点. 由于在搜索过程中对每个顶点一次必须访问所有后继，因此称为广度优先搜索. 形象地说，广度优先搜索可比喻为“由近及远，齐向各方”.

以 T 表队列，开始时 $T=\varnothing$，即 T 是空队列. 令

$$l(v)=\begin{cases}0, & \text{顶点 } v \text{ 未访问},\\ 1, & \text{顶点 } v \text{ 已访问}.\end{cases}$$

第 1 步　若所有顶点都被访问，即

$$l(v)=1,\ v\in V(G),$$

终止. 否则取未访问点 v，即 $l(v)=0$. 访问 v，v 进入队列 T，令 $l(v)=1$.

第 2 步　队列 T 的队头 v 退出队列.

第 3 步　若 $\Gamma(v)=\varnothing$ 或 $l(w)=1,\ w\in\Gamma(v)$，

转第 4 步，否则转第 5 步.

第 4 步　若队列成为空队列，转第 1 步，否则转第 2 步.

第 5 步　对所有 w：$w\in\Gamma(v)$ 且 $l(w)=0$，访问 w，即令 $l(w)=1$，w 进入队列 T，转第 4 步.

例 6.3　用算法 6.1 遍历图 6.1(a) 的图 G.

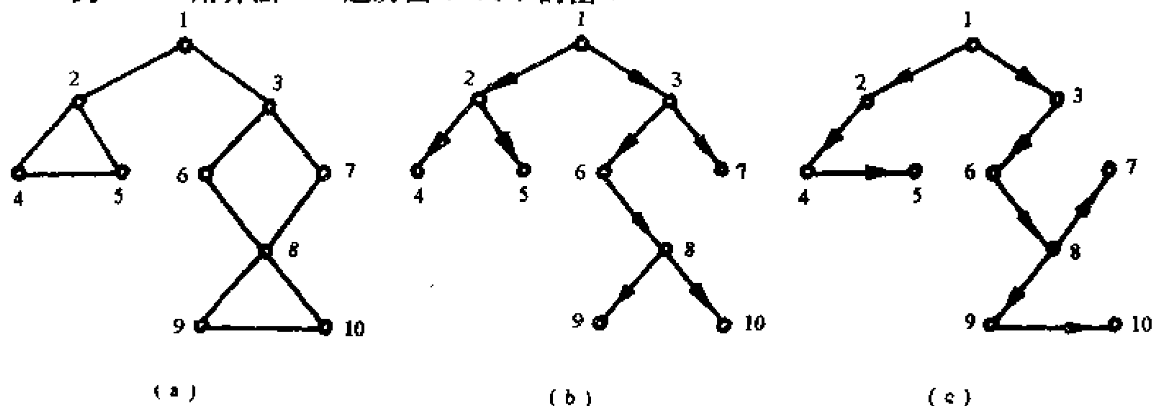


图 6.1　(a) 图 G　(b) 广度优先搜索过程　(c) 深度优先搜索过程

首先用后继表法列出每个顶点的邻域表如下：

1：2,3	2：1,4,5
3：1,6,7	4：2,5
5：2,4	6：3,7,8
7：3,6,8	8：6,7,9,10
9：8,10	10：8,9

选顶点 1 开始访问，依次访问的过程如下：

(1) 访问顶点 1 的所有相邻顶点 2 和 3，

(2) 访问顶点 2 的未被访问的相邻顶点 4 与 5，

(3) 访问顶点 3 的未被访问的相邻顶点 6 和 7，

(4) 顶点 4 与 5 的相邻顶点都已访问，

(5) 访问顶点 6 的未被访问的相邻顶点 8，

(6) 顶点 7 的相邻顶点都已访问，

(7) 访问顶点 8 的未被访问的相邻顶点 9 与 10，

(8) 顶点 9 与 10 的相邻顶点都已访问，

(9) 队列已成为空队列，

(10) 所有顶点都被访问，终止.

其遍历过程如图 6.1(b). 由此得到的访问序列是

$$v_1 \to v_2 \to v_3 \to v_4 \to v_5 \to v_6 \to v_7 \to v_8 \to v_9 \to v_{10}.$$

算法 6.2　遍历图的深度优先搜索法

假设从访问顶点 v_1 开始，然后访问 v_1 的任意未被访问的一个后继，再从这个顶点出发，访问其未被访问的一个后继，继续这一访问过程，当访问到某个顶点 v_i 而 v_i 没有后继或者 v_i 的后继都被访问，则逆向退回到 v_i 的前趋，继续新的访问，直到图中所有从 v_1 可达的顶点都被访问. 此时，若图的顶点都被访问，则搜索终止，否则选择一个未被访问的顶点，重复上述过程，直到图中所有顶点都被访问时迭代终止. 由于在搜索过程中每次选定一个方向使访问的顶点尽可能延伸，当访问不能进行时则返回，再继续新的访问，因此称为深度优先搜索. 形象地说，深度优先搜索可比喻为“直往前行，碰壁回头”.

用标号 l 表示顶点是否被访问，即

$$l(v) = \begin{cases} 0, & \text{顶点 } v \text{ 未访问}, \\ 1, & \text{顶点 } v \text{ 已访问}. \end{cases}$$

再用标号 p 表示深度优先搜索的顺序，若

$$p(w) = v,$$

则表示在访问顶点 v 之后紧接着访问顶点 w，即有一条有向边 (v,w)，或者说 v 是 w 的父顶点，w 是 v 的子顶点.

第 1 步　取 $v \in V(G)$, $l(v) = 0$.

访问 v，即令 $l(v) = 1$.

第 2 步　若 $\Gamma(v) = \varnothing$ 或

$$l(w) = 1,\ w \in \Gamma(v),$$

转第 3 步，否则转第 4 步.

第 3 步　若所有顶点都被访问，即

$$l(v) = 1,\ v \in V(G),$$

终止. 否则，若 v 没有父顶点，转第 1 步；不然设 v 的父顶点是 u，令

$$v = u,$$

转第 2 步.

第 4 步　取 $w \in \Gamma(v)$ 且 $l(v) = 0$.

访问 w，即令 $l(w) = 1$.

w 是 v 的子顶点，即令 $p(w) = v$.

第 5 步　令 $v = w$，转第 2 步.

例 6.4　用算法 6.2 遍历图 6.1(a) 的图 G.

选顶点 1 开始访问，依次访问过程如下：

(1) 访问 1 的后继 2，

(2) 访问 2 的后继 4，

(3) 访问 4 的后继 5，

(4) 顶点 5 的相邻顶点都已访问，退回到父顶点 4，再退回到顶点 2，再退回到顶点 1，

(5) 访问 3，再访问 6,8,9,10，

(6) 顶点 10 的相邻顶点都已访问，退回到父顶点 9，再退回到顶点 8，

(7) 访问 7，

(8) 退回到顶点 8，再依次退回到顶点 6,3,1. 所有顶点都被访问，终止.

其遍历过程如图 6.1(c). 由此得到的访问序列是

$$v_1 \to v_2 \to v_4 \to v_5 \to v_3 \to v_6 \to v_8 \to v_9 \to v_{10} \to v_7.$$

图的遍历过程实质上是查找每个顶点的相邻顶点，对 n 个顶点的图，每个顶点至多与其余 $n-1$ 个顶点相邻，故执行时间与 $n(n-1)$ 成比例，因此图的遍历的两种算法都是好算法.

§6.4　连通性算法

对一个图 G，常常要问的第一个问题就是 G 是否连通，若不连退，有几个连通支.

关于连通性，在 §1.8 指出，图 G 有 ω 个连通支的充要条件是，图 G 的邻接矩阵 H 或关联矩阵 A 在适当安排顶点和边的顺序后可写为有 ω 个块的分块对角形矩阵. 对有 n 个顶点 m 条边的图，顶点的顺序有 $n!$ 种，边的顺序有 $m!$ 种，因此，穷举法判断连通性是指数算法.

利用推论 1.19，对 n 个顶点($n > 2$) 的图 G，设 H 是邻接矩阵，则 G 连通的充要条件是

$$I + H + H^2 + \cdots + H^{n-1} > 0.$$

这个方法比穷举法有效，但包含大量矩阵的乘法，仍是不可取的.

判断图的连通性的简单而有效的算法是融合顶点法.

定义 6.4　设$(u,v) \in E(G)$，若用一个新顶点 w 代替顶点 u 与 v，而与顶点 u 和 / 或 v 关联的边都与新顶点 w 关联，则称为**融合**顶点 u 与 v，并称 w 为**融合点**.

由定义，一次融合顶点数减少 1，边数不变.

融合顶点法的基本步骤是任选图 G 的一个顶点 v，融合与顶点相邻的所有顶点得融合点，不妨仍记为 v，再融合与顶点 v 相邻的所有顶点，重复这一过程，直到图 G 没有顶点能融合到 v 为止. 此时，若 G 的所有顶点都被融合，则 G 是连通的，否则选择图 G 的未被融合的顶点，重新开始融合过程. 当融合不能进行时终止.

容易看到，当过程终止时，有多少个融合点图 G 就有多少个连通支. 每个融合点所包含的图 G 的顶点集的导出子图就是 G 的连通支.

例 6.5 在图 6.1(a) 的图 G，选择顶点 1，融合相邻顶点 2 与 3，如图 6.2(a)，为简单，在融合后的图中略去环和平行边，并把融合点仍记为 1. 再融合所有相邻顶点 4,5,6 与 7，如图 6.2(b). 再融合顶点 8，如图 6.2(c). 再融合顶点 9 与 10，如图 6.2(d). 因为图 G 的顶点最后全部在一个融合点中，故 G 连通.

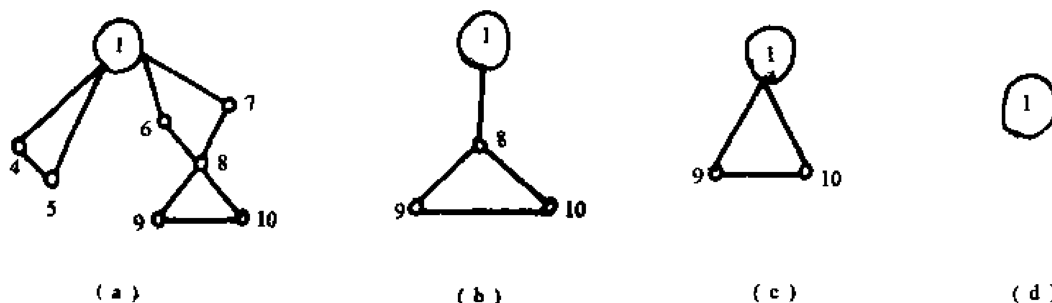


图 6.2　融合顶点过程，大圆圈表融合顶点

两个顶点的融合可在图 G 的邻接矩阵 H 上进行. 如果融合顶点 i 与 j，而融合点仍记为 i，且略去平行边，则把 j 行加到 i 行，再把 j 列加到 i 列，然后去掉 j 行 j 列. 这里的加是逻辑加"$\vee$"，即

$$1 \vee 0 = 0 \vee 1 = 1 \vee 1 = 1,$$
$$0 \vee 0 = 0.$$

对一个图 G 的融合，融合次数最多 $|V(G)| - 1$ 次，而每次融合最多 $|V(G)|$ 次逻辑加，因此融合顶点法的执行时间与 $|V(G)|(|V(G)| - 1)$ 成比例，故这是一个好算法.

算法 6.3　连通性的融合顶点法

设 G 是 (n,m) 图，用 k 表连通支数.

第 1 步　输入邻接矩阵 H，$k = 1$.

第 2 步　选未被记录的顶点 $i \in V(G)$.

第 3 步　在 H 中检查 i 列，若没有与顶点 i 相邻的顶点，转第 4 步，否则转第 6 步.

第 4 步　记录由顶点 i 产生的连通支.

第 5 步　若没有未记录的顶点，输出连通支数和连通支，停. 否则令 $k = k + 1$，转第 2 步.

第 6 步　记录与顶点 i 相邻的所有顶点，把相应的行加于 i 行，相应的列加于 i 列，然后在 H 中置与 i 相邻的所有顶点的行、列为零，转第 3 步.

例 6.6 用算法 6.3 求图 6.1(a) 的图的连通性.

$$
\text{邻接矩阵} \quad H = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix},
$$

选未被记录的顶点 1,检查 H 的第 1 列,有顶点 2 与 3 相邻接. 把第 2 行与第 3 行加于第 1 行,再把第 2 列与第 3 列加于第 1 列,然后置第 2 行第 3 行与第 2 列第 3 列为零,得

$$
H = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}, \quad \text{(参看图 6.2}(a)\text{)}
$$

记录与 1 相邻的顶点 2 与 3.

在 H 中检查到顶点 1 的相邻顶点 4,5,6 与 7. 把第 4,5,6 与 7 行加于第 1 行,第 4,5,6 与 7 列加于第 1 列,并置第 4,5,6 与 7 行和相应列为零,得

$$
H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}, \quad \text{(参看图 6.2}(b)\text{)}
$$

记录与 1 相邻的顶点 4,5,6 与 7.

在 H 中顶点 1 与 8 相邻,把第 8 行加于第 1 行,第 8 列加于第 1 列,并置第 8 行与第 8

列为零，得

$$H=\begin{bmatrix}1&0&0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0&0&1\\1&0&0&0&0&0&0&0&1&0\end{bmatrix},\quad \text{(参看图 6.2}(c))$$

记录与顶点 1 相邻的顶点 8.

在 H 中顶点 1 与顶点 8,9 相邻，把第 8,9 行加于第 1 行，第 8,9 列加于第 1 列，并置第 8,9 行与第 8,9 列为零，得

$$H=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix},\quad \text{(参看图 6.2}(d))$$

记录与 1 相邻的顶点 8 与 9.

因为 H 中检查出顶点 1 没有相邻顶点，且 G 的所有顶点都已记录，故 G 只有一个连通支，即 G 是连通的.

§6.5 强连通性算法

融合顶点法不能用来判定有向图的强连通性.

设 G 是 (n,m) 有向图，P 是可达矩阵，则两个顶点 i 与 j 同属于一个强连通支的充要条件是 $p_{ij}=p_{ji}=1$，等价地，即

$$p_{ij}\wedge p_{ji}=1. \tag{6.1}$$

这里“∧”是逻辑乘，规则是

$$0\wedge 0=0\wedge 1=1\wedge 0=0,$$
$$1\wedge 1=1.$$

因此，令

$$m_{ij} = m_{ji} = \begin{cases} 0, & p_{ij} \wedge p_{ji} = 0, \\ 1, & p_{ij} \wedge p_{ji} = 1, \end{cases} \tag{6.2}$$

则 $M = (m_{ij})$ 是对称阵，且第 i 行元素 $m_{i1}, m_{i2}, \cdots, m_{in}$ 中的非零元素所在列相应的顶点子集的导出子图就是 G 的一个强连通支.

现在的问题是如何求可达矩阵 P. 在 §1.8 中给出了利用邻接矩阵求可达矩阵的方法，由于这一方法包含了大量矩阵的乘法，因此不是有效的.

换一个角度来考虑. 如果把"可达"作为两个顶点间的"关系"，则可达关系是传递关系. 设 H 是邻接矩阵，若 H 的 1 列 j 行处的元是"1"，表示顶点 1 是顶点 j 的后继，即顶点 j 可达顶点 1. 如果把 H 的第 1 行加到第 j 行(逻辑加)，则表示凡顶点 1 可达的，顶点 j 也一定可达. 因此，若依次考查 k 列，$k = 1, 2, \cdots, n$，则由逻辑加运算即可求得可达矩阵.

算法 6.4 求可达矩阵的逻辑算法

设 G 是 (n, m) 有向图.

第 1 步 输入邻接矩阵 H，$i = 1$.

第 2 步 对 j，$1 \leqslant j \leqslant n$，$j \neq i$，若 $H[j,i] = 1$，则令

$$H[j,k] = H[j,k] \vee H[i,k],\ k = 1, 2, \cdots, n.$$

第 3 步 若 $i \leqslant n-1$，则令 $i = i + 1$，转第 2 步.

第 4 步 令 $P = H$，再令

$$p_{ii} = 1,\ i = 1, 2, \cdots, n,$$

则 P 为可达矩阵.

由于解决了求可达矩阵的问题，因此也就解决了求强连通性的问题.

算法 6.5 强连通性的逻辑算法

设 G 是 (n, m) 有向图.

第 1 步 求可达矩阵 P.

第 2 步 令 $M = (m_{ij})$，其中 m_{ij} 由式(6.2)计算.

第 3 步 若 M 的 i 行的非零元在 $j_1, j_2, \cdots, j_r$ 列，则导出子图

$$G(\{j_1, j_2, \cdots, j_r\})$$

就是 G 的强连通支.

输出强连通支数和各强连通支，停.

在强连通性算法中，对顶点 i，邻接矩阵的 i 行至多与另外 $(n-1)$ 行作逻辑运算，即至多作 $n(n-1)$ 次逻辑加. 因此，执行时间与 $n^2(n-1)$ 成比例.

对无向图，顶点 j 与顶点 i 相邻，可看作顶点 j 是顶点 i 的后继，因此，算法 6.5 对无向图也是适用的.

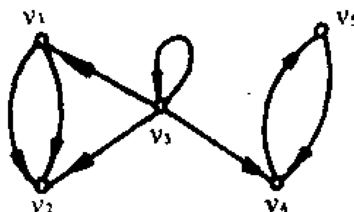


图 6.3

例 6.7 求图 6.3 的有向图的强连通性.

邻接矩阵

$$H=\begin{pmatrix}0&1&0&0&0\\0&0&1&0&0\\1&0&1&1&0\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix}.$$

先求可达矩阵 P，由

$$H=\begin{pmatrix}0&1&0&0&0\\0&0&1&0&0\\1&0&1&1&0\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix}\xrightarrow[\text{第 3 行}]{\substack{i=1\\ \text{第 1 行加到}}}\begin{pmatrix}0&1&0&0&0\\0&0&1&0&0\\1&1&1&1&0\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix}\xrightarrow[\text{第 1,3 行}]{\substack{i=2\\ \text{第 2 行加到}}}$$

$$\begin{pmatrix}0&1&1&0&0\\0&0&1&0&0\\1&1&1&1&0\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix}\xrightarrow[\text{第 1,2 行}]{\substack{i=3\\ \text{第 3 行加到}}}\begin{pmatrix}1&1&1&1&0\\1&1&1&1&0\\1&1&1&1&0\\0&0&0&0&1\\0&0&0&1&0\end{pmatrix}\xrightarrow[\text{第 1,2,3,5 行}]{\substack{i=4\\ \text{第 4 行加到}}}\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&0&1\\0&0&0&1&1\end{pmatrix}$$

$$\xrightarrow[\text{第 1,2,3,4 行}]{\substack{i=5\\ \text{第 5 行加到}}}\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix},$$

∴　　可达矩阵

$$P=\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix}.$$

再由式(6.2)求矩阵 M：

$$M=\begin{pmatrix}1&1&1&0&0\\1&1&1&0&0\\1&1&1&0&0\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix}.$$

∴　　有 2 个强连通支：

$$G(\{v_1,v_2,v_3\}),\ G(\{v_4,v_5\}).$$

对于有向图的强连通性可利用深度优先搜索树，读者自己给出计算步骤(习题 6.34 与 6.35)。

§6.6　求生成树(林)

求图的生成树(林)算法，对连通图是求生成树，对不连通图是求生成林. 反过来，若

求得生成树，图是连通的，若求得生成林，图是不连通的.

设 G 是 (n,m) 连通图，今从 G 中任选 $(n-1)$ 条边，若无圈或连通，则由定理 2.1，得 G 的一棵生成树. 这种方法需要 $\binom{m}{n-1}$ 次判断是否无圈或是否连通，实际上这就是穷举法.

求生成树(林)的一个好的算法是利用如下性质：对连通图，其极大连通子图是生成树，对不连通图，其极大连通子图构成生成林. 将图的边编号，依次检查每一条边. 当正在检查的边与已标记的所有边不含圈时，该边生长到树中并作标记，然后检查下一条边；若含圈，则不作标记，并继续检查下一条边. 最后得到作标记的边数是 $(n-c)$，当 $c=1$ 时是生成树，当 $c \geqslant 2$ 时是生成林. 由于每次生长一条边在树(林)中，故称为边生长算法.

用二数组法输入图 G：$D_1(I)$，$D_2(I)$. c 表已经得到的树的数目. 设

$$V(i)=\begin{cases}0, & \text{顶点 } i \text{ 不在已经得到的树中}, \\ k, & \text{顶点 } i \text{ 在树 } T_k \text{ 中}.\end{cases}$$

$$E(j)=\begin{cases}0, & \text{边 } j \text{ 不在已经得到的树中}, \\ k, & \text{边 } j \text{ 在树 } T_k \text{ 中}.\end{cases}$$

今考查边 $k=(D_1(k),D_2(k))$，有五种情形.

情形 1　$D_1(k)$，$D_2(k)$ 都不在已经得到的树中，即

$$D_1(k)=D_2(k)=0. \tag{6.3}$$

此时，边 k 加入新的一棵树 T_{c+1} 中.

情形 2　$D_1(k)$ 在 T_i 中，$D_2(k)$ 在 T_j 中，$i \neq j$，即

$$V(D_1(k))=i,\ V(D_2(k))=j,\ i \neq j. \tag{6.4}$$

此时边 k 连接 T_i 与 T_j 为一棵树，所得的树减少一棵.

情形 3　$D_1(k)$，$D_2(k)$ 都在 T_i 中，即

$$V(D_1(k))=V(D_2(k))=i. \tag{6.5}$$

因为 T_i 是树，故附加边 $k=(D_1(k),D_2(k))$ 之后有唯一的圈，所以这样的边不能生长到树中.

情形 4　$D_1(k)$ 在 T_i 中，$T_2(k)$ 不在已经得到的树中，即

$$V(D_1(k))=i,\ V(D_2(k))=0. \tag{6.6}$$

此时，边 k 生长到 T_i 中.

情形 5　$D_1(k)$ 不在已经得到的树中，$D_2(k)$ 在 T_j 中，即

$$V(D_1(k))=0,\ V(D_2(k))=j. \tag{6.7}$$

此时，边 k 生长到 T_j 中.

由于此算法依次考查 G 的 m 条边，故执行时间与 m 成比例.

算法 6.6　边生长算法

设 G 是 (n,m) 图. 在开始，

二数组 $D_1(k), D_2(k), k = 1,2,\cdots,m$,

$V(i) = 0,\ i = 1,2,\cdots,n$,

$E(j) = 0,\ j = 1,2,\cdots,m$,

$k = 1,\ c = 0,\ r = 0.$

第 1 步　若 $V(D_1(k)) = 0$,转第 2 步,

若 $V(D_1(k)) = i \neq 0$,转第 3 步.

第 2 步　若 $V(D_2(k)) = 0$,转第 6 步,

若 $V(D_2(k)) = j \neq 0$,转第 7 步.

第 3 步　若 $V(D_2(k)) = 0$,转第 4 步,

若 $V(D_2(k)) = j \neq 0$,转第 5 步.

第 4 步　即情形 4,边 k 生长到 T_i 中,即令

$$T_i = T_i + k,$$

$$E(k) = V(D_2(k)),$$

转第 9 步.

第 5 步　若 $i = j$ 即情形 3,转第 10 步,否则转第 8 步.

第 6 步　即情形 1,增加一棵树,即令

$$c = c + 1,$$

$$T_c = [k],$$

$$E(k) = V(D_1(k)) = V(D_2(k)) = c,$$

转第 9 步.

第 7 步　即情形 5,边 k 生长到 T_j 中,即令

$$T_j = T_j + k,$$

$$E(k) = V(D_1(k)) = j,$$

转第 9 步.

第 8 步　即情形 2,边 k 连接 T_i 与 T_j 为一棵树.

(1) 当 $i < j$,把边 k 和 T_j 合并到 T_i,即令

$$T_i = T_i + T_j + k,$$

$$V(t) = \begin{cases} V(t) & ,\text{当 } V(t) < j, \\ i & ,\text{当 } V(t) = j, \\ V(t) - 1 & ,\text{当 } V(t) > j, \end{cases}$$

$$t = 1,2,\cdots,n.$$

$$E(t) = \begin{cases} E(t) & ,\text{当 } E(t) < j, \\ i & ,\text{当 } E(t) = j \text{ 或 } t = k, \\ E(t) - 1 & ,\text{当 } E(t) > j, \end{cases}$$

$$t = 1,2,\cdots,m.$$

当 $i > j$,把边 k 和 T_i 合并到 T_j,即令

$$T_j = T_i + T_j + k,$$

$$V(t) = \begin{cases} V(t) & ,\text{当 } V(t) < i, \\ j & ,\text{当 } V(t) = i, \\ V(t) - 1 & ,\text{当 } V(t) > i, \end{cases}$$

$t = 1, 2, \cdots, n.$

$$E(t) = \begin{cases} E(t) & ,\text{当 } E(t) < i, \\ j & ,\text{当 } E(t) = i \text{ 或 } t = k, \\ E(t) - 1 & ,\text{当 } E(t) > i, \end{cases}$$

$t = 1, 2, \cdots, m.$

(2) 令 $c = c - 1$,转第 9 步.

第 9 步　生长到树中的边增加一条,故令

$$r = r + 1.$$

第 10 步　若 $r = n - 1$ 或 $k = m$,则输出 $T_1, T_2, \cdots, T_c$.停.否则令 $k = k + 1$,转第 1 步.

注意到在第 10 步增加了一个控制条件 $r = n - 1$,其中 r 表生长到树中的边数.由于生成树的边正好 $(n - 1)$ 条,而生成林的边少于 $(n - 1)$ 条,因此当 $m >> n$ 时,不必检查图 G 的所有 m 条边,最多只需生长 $(n - 1)$ 条边即可终止,从而大为减少计算量.

算法 6.6 是依次检查图 G 的每条边,也可按遍历图 G 的顶点的方法求生成树(林).

设想从图 G 的某一顶点开始旅行,旅行的规则是,沿没有旅行过的边到达没有到达过的顶点.这样的旅行自然不会形成圈,因此,当旅行终止时便得到图 G 的初始旅行点所在连通支的极大无圈生成子图,即该连通支的生成树.如果图 G 连通,算法终止,如果图 G 不连通,则另选顶点开始新的旅行.总之,一个图有多少个连通支就需要多少次遍历过程.遍历有两种方法,由广度优先搜索法得到的称为**广度优先生成树**(林),由深度优先搜索法得到的称为**深度优先生成树**(林)

算法 6.7　深度优先生成树算法

设 G 连通,按后继表法存贮图 G,T 表当前得到的树,$l(i)$ 表顶点 i 生长到 T 中的顺序号,任取顶点 v 开始,遍历过程终止时得到以 v 为根的外向树.因为所得到的图是有向图,所以称树 T 的边为**正向边**,称余树 S 的边为**返回边**.在开始

$$T = \varnothing,\ S = \varnothing,\ l(u) = 0\ \forall\ u \in V(G).$$

第 1 步　取 $v \in V(G)$,令 $l(v) = 1, k = 1$.

第 2 步　取顶点 v 的第 1 个未检查的边,设为 (v,w),并规定方向为从 v 到 w,转第 3 步.若顶点 v 的关联边都已检查,转第 4 步.

第 3 步　若 $l(w) = 0$,令

$$T = T + (v,w),\ v = w,$$

$$k = k + 1,\ l(w) = k,$$

转第 2 步.若 $l(w) \neq 0$,令 $S = S + (v,w)$,转第 2 步

第 4 步　若有边 $(u,v) \in T$,令 $v = u$,转第 2 步,否则输出深度优先生成树 T,停.

例 6.8 求图 6.1(a) 的图 G 的深度优先生成树.

如图 6.4，顶点旁边的数字表生长到 T 的顺序号. 在第 3 步，若 $l(w) = 0$，则 (v,w) 是正向边，用实线表示；若 $l(w) \neq 0$，则 (v,w) 是返回边，用虚线表示. 由深度优先搜索过程，正向边由小顺序号指向大顺序号，返回边由大顺序号指向小顺序号.

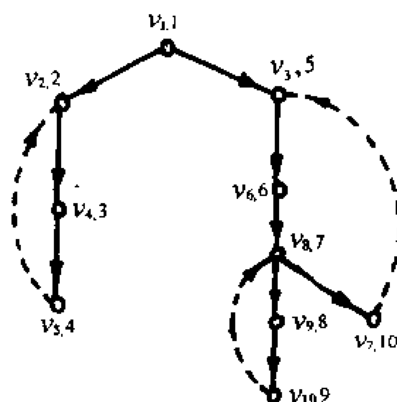


图 6.4 深度优先生成树

算法 6.8 广度优先生成树算法

设 G 连通，按后继表法存贮图 G，T 表当前得到的树，S 表不在 T 的边，R 表队列，$l(i)$ 表顶点 i 生长到 T 中的顺序号. 任取顶点 v 开始，遍历过程终止时得到以 v 为根的外向树. 在开始

$$T = \varnothing,\ S = \varnothing,\ R = \varnothing,\ l(u) = 0\ \forall\ u \in V(G).$$

第 1 步　取 $v \in V(G)$，令 $l(v) = 1, k = 1$.

第 2 步　取顶点 v 的第 1 个未检查的边，设为 (v,w)，并规定方向为从 v 到 w，转第 3 步. 若顶点 v 的关联边都已检查，转第 4 步.

第 3 步　若 $l(w) = 0$，令

$$T = T + (v,w),\ R = R \cup \{w\},$$
$$k = k + 1, l(w) = k,$$

转第 2 步. 若 $l(w) \neq 0$，令

$$S = S + (v,w),$$

转第 2 步.

第 4 步　若 $R = \varnothing$，输出广度优先生成树 T，停. 否则转第 5 步.

第 5 步　取 R 的第 1 个元为 v，并令 $R = R\backslash\{v\}$，转第 2 步.

例 6.9 求图 6.1(a) 的图 G 的广度优先生成树.

如图 6.5，顶点旁的数字表示生长顺序号，实线表树枝，虚线表连枝. 树枝的方向由小顺序号指向大顺序号，但连枝的方向不一定是从大顺序号指向小顺序号.

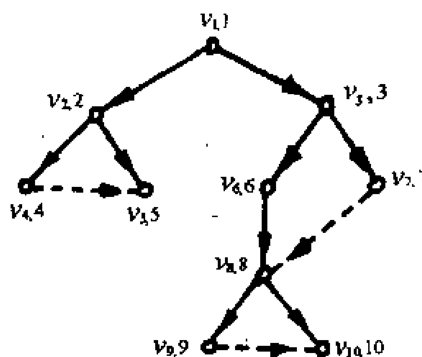


图 6.5 广度优先生成树

§6.7 求全部生成树

假设 G 是 (n,m) 连通图，任取 $e \in E(G)$，则 G 的生成树分为两类，第一类为含 e 的生成树，第二类为不含 e 的生成树. 作 G 的两个子图 G_1 与 G_2：G_1 是把 G 的边 e 作标记的图，G_2 是

G 去掉边 e 的图. 于是,第一类生成树都在 G_1 中且边 e 作了标记,第二类生成树都在 G_2 中. 对 G_1 与 G_2,各自又可分为两个子图. 继续进行,当出现全由标记边组成的圈或生成树时,这个图就处理完毕,然后这个图就被删去. 当所有分解出的图都处理完毕时,全部生成树就得到了. 这就是求全部生成树的基本思想.

按照深度优先搜索的原则,对各层次中分解出的两个图 G_1 与 G_2,总可先处理 G_1 而把 G_2 放入栈. 当进行到某一层次 G_1 处理完毕时,即按栈的后进先出原则,转而处理最后进入栈的 G_2. 直到栈的元素全部处理完毕,深度优先搜索过程终止,得到全部生成树.

由定理 2.7,若 e 是图 G 的桥,则 G 的所有生成树都含 e. 因此,不妨假定 G 没有桥,或者说,当 G 有桥 e 时,按定义 5.8,对边 e 缩并,然后再处理缩并图 $G\cdot e$.

注意到在每一层次的 G_2 都是去掉 G 的一条边得到的,因此 G_2 有可能不连通,而不连通的图是不含生成树的,所以当 G_2 不连通时不必入栈而把它删去. 考虑到判定各层次的 G_2 是否连通需花大量工作,故可采用一个简单的办法,如果 $|E(G_2)| < |V(G)| - 1$,则 G_2 不会是连通的而予以删去.

采用二数组法存贮图 G:$D_1(I)$,$D_2(I)$. 用一维数组 $E(i)$ 描述正在检验中的图 G 的边:

$$E(i)=\begin{cases}0 & ,\text{边 } i \text{ 不在 } G \text{ 中},\\ 1 & ,\text{边 } i \text{ 在 } G \text{ 中但未标记},\\ -1 & ,\text{边 } i \text{ 在 } G \text{ 中且已标记}.\end{cases}$$

用一维数组 $V(j)$ 描述正在处理的子图中标记边形成的连通支的分布情况:

$$V(j)=\begin{cases}0, & \text{顶点 } j \text{ 不在任何标记边的连通支内},\\ i, & \text{顶点 } j \text{ 在标记边的第 } i \text{ 个连通支内}.\end{cases}$$

这样,当取边 i 作标记生成 G_1 时,只需检验边 i 的两个端点是否在原有标记边的同一个连通支内,即是否有

$$V(D_1(i)) = V(D_2(i)),$$

从而判断当边 i 标记后是否由当前标记边构成圈.

栈由两个二维数组 $VS(IT,J)$ 与 $ES(IT,I)$ 表示,$j=1,2,\cdots,n$,$i=1,2,\cdots,m$,其中 IT 表栈所在的层次,$VS(IT,J)$ 是顶点栈,与 $V(J)$ 对应,$ES(IT,I)$ 是边栈,与 $E(I)$ 对应.

今检验边 e_i,先生成 G_2,即令 $E(i)=0$. 当 $|E(G_2)|<n-1$ 时删去 G_2,否则 G_2 入栈. 然后生成 G_1,即令 $E(i)=-1$,有五种情形.

情形 1 $V(D_1(i))=V(D_2(i))=0$,即边 i 的两个端点不是任何标记边的端点,此时边 i 加入新的连通支.

情形 2 $V(D_1(i))=0$,$V(D_2(i))\neq 0$,即顶点 $D_1(i)$ 不在标记边的连通支内,故边 i 加入顶点 $D_2(i)$ 所在连通支.

情形 3 $V(D_1(i))\neq 0$,$V(D_2(i))=0$,与情形 2 类似,边 i 加入顶点 $D_1(i)$ 所在连通支.

情形 4 $V(D_1(i))=V(D_2(i))\neq 0$,即边 i 的两个端点 $D_1(i)$ 与 $D_2(i)$ 在同一连通支,加入边 i 后必构成圈,故删去此图,并从栈顶退出一个子图继续检验.

情形 5 $V(D_1(i))\neq 0$,$V(D_2(i))\neq 0$,且它们不相等,即边 i 的两个端点在不同的连通支,当标记边 i 后,边 i 连接两个连通支为一个连通支. 合并后的连通支的顶点标号取合并前两个连通支的小标号. 此时连通支减少一个.

在情形 2 或 3 或 5 发生时，连通支的标记边都增加了一条，由于合并后的连通支取合并前的小标号，故当

$$V(k) = 1,\ k = 1,2,\cdots,n$$

时便得到一棵生成树.

算法 6.9 求全部生成树的深度优先搜索法

设 G 是 (n,m) 图，用二数组法存贮图 G：$D_1(I)$，$D_2(I)$。在开始，正在检验的图是给定的图 G，即令

$$E(i) = 1,\ i = 1,2,\cdots,m,$$
$$V(j) = 0,\ j = 1,2,\cdots,n,$$
$IT = 1$，即栈是空栈，
$IS = 1$，即没有连通支，
$NT = 0$，即没有生成树.

第 1 步　取边 e_i，i 满足

$$i = \min\{k \mid E(k) = 1,\ k = 1,2,\cdots,n\}.$$

第 2 步　生成 G_2，即令 $E(i) = 0$.

若 $|E(G_2)| < n - 1$，转第 3 步，否则入栈，即令

$$ES(IT,k) = E(k),\ k = 1,2,\cdots,m,$$
$$VS(IT,k) = V(k),\ k = 1,2,\cdots,n,$$
$$IT = IT + 1.$$

第 3 步　生成 G_1，即令 $E(i) = -1$.

第 4 步　若 $V(D_1(i)) = 0$，转第 5 步，否则转第 6 步.

第 5 步　若 $V(D_2(i)) = 0$，转第 7 步，否则转第 8 步.

第 6 步　若 $V(D_2(i)) = 0$，转第 9 步，否则转第 10 步.

第 7 步　即情形 1，增加一个连通支，即令

$$V(D_1(i)) = V(D_2(i)) = IS,$$
$$IS = IS + 1,$$

转第 1 步.

第 8 步　即情形 2，令 $V(D_1(i)) = V(D_2(i))$，转第 12 步.

第 9 步　即情形 3，令 $V(D_2(i)) = V(D_1(i))$，转第 12 步.

第 10 步　若 $V(D_1(i)) \neq V(D_2(i))$，转第 11 步，否则转第 13 步.

第 11 步　即情形 5，边 i 连接 $D_1(i)$ 所在连通支与 $D_2(i)$ 所在连通支为一个连通支.

(1) 对 $k = 1,2,\cdots,n$，若 $V(D_1(i)) < V(D_2(i))$，则

$$V(k) = \begin{cases} V(k) & ,\ \text{当 } V(k) < V(D_2(i)), \\ V(D_1(i)) & ,\ \text{当 } V(k) = V(D_2(i)), \\ V(k) - 1 & ,\ \text{当 } V(k) > V(D_2(i)). \end{cases}$$

若 $V(D_1(i)) > V(D_2(i))$，则

$$V(k)=\begin{cases}V(k) &, 当 V(k)<V(D_1(i)),\\ V(D_2(i)) &, 当 V(k)=V(D_1(i)),\\ V(k)-1 &, 当 V(k)>V(D_1(i)).\end{cases}$$

(2) 令 $IS=IS-1$.

第 12 步　若 $V(k)=1, k=1,2,\cdots,n$,则令

$$NT=NT+1,$$

输出第 NT 棵生成树,即边集

$$\{e_i|E(i)=-1, i=1,2,\cdots,m\},$$

转第 13 步,否则转第 1 步.

第 13 步　若 $IT=1$,即栈已空,已经得到全部生成树,停.

第 14 步　此时或者是情形 4 形成圈,或者是情形 2、3、5 之后得到生成树,该图处理完毕,退栈,即令

$$IT=IT-1,$$
$$E(k)=ES(IT,k),\ k=1,2,\cdots,m,$$
$$V(k)=VS(IT,k),\ k=1,2,\cdots,n,$$
$$IS=1+\max\{V(k)|k=1,2,\cdots,n\}.$$

转第 1 步.

例 6.10　求图 6.6(a) 的全部生成树.

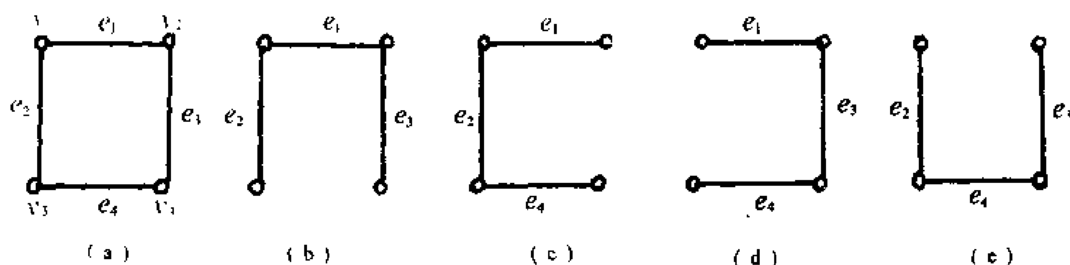


图 6.6　(a) 图 G　(b)～(e) 全部生成树

开始:　$D_1=(1,1,2,3)$, $D_2=(2,3,4,4)$

$E=(1,1,1,1)$, $V=(0,0,0,0)$

$IT=1$, $IS=1$, $NT=0$

1. 取 e_1.

2. $ES(1)=(0,1,1,1)$, $VS(1)=(0,0,0,0)$, $IT=2$.

3. $E=(-1,1,1,1)$.

7. $V=(1,1,0,0)$, $IS=2$.

1. 取 e_2.

2. $ES(2)=(-1,0,1,1)$, $VS(2)=(1,1,0,0)$, $IT=3$.

3. $E=(-1,-1,1,1)$.

9. $V=(1,1,1,0)$.

1. 取 e_3.

2. $ES(3) = (-1, -1, 0, 1)$, $VS(3) = (1, 1, 1, 0)$, $IT = 4$.

3. $E = (-1, -1, -1, 1)$.

9. $V = (1, 1, 1, 1)$.

12. $NT = 1$,第 1 棵生成树$[e_1, e_2, e_3]$,如图 6.6(b).

14. $IT = 3$, $E = ES(3) = (-1, -1, 0, 1)$,

$V = VS(3) = (1, 1, 1, 0)$, $IS = 2$.

1. 取 e_4.

2. $E = (-1, -1, 0, 0)$,删去.

3. $E = (-1, -1, 0, -1)$.

9. $V = (1, 1, 1, 1)$.

12. $NT = 2$,第 2 棵生成树$[e_1, e_2, e_4]$,如图 6.6(c).

14. $IT = 2$, $E = ES(2) = (-1, 0, 1, 1)$.

$V = VS(2) = (1, 1, 0, 0)$, $IS = 2$.

1. 取 e_3.

2. $E = (-1, 0, 0, 1)$,删去.

3. $E = (-1, 0, -1, 1)$.

9. $V = (1, 1, 0, 1)$.

1. 取 e_4.

2. $E = (-1, 0, -1, 0)$,删去.

3. $E = (-1, 0, -1, -1)$.

8. $V = (1, 1, 1, 1)$.

12. $NT = 3$,第 3 棵生成树$[e_1, e_3, e_4]$,如图 6.6(d).

14. $IT = 1$, $E = ES(1) = (0, 1, 1, 1)$,

$V = VS(1) = (0, 0, 0, 0)$, $IS = 1$.

1. 取 e_2.

2. $E = (0, 0, 1, 1)$,删去.

3. $E = (0, -1, 1, 1)$.

7. $V = (1, 0, 1, 0)$, $IS = 2$.

1. 取 e_3.

2. $E = (0, -1, 0, 1)$,删去.

3. $E = (0, -1, -1, 1)$.

7. $V = (1, 2, 1, 2)$, $IS = 3$.

1. 取 e_4.

2. $E = (0, -1, -1, 0)$,删去.

3. $E = (0, -1, -1, -1)$.

11. $V = (1, 1, 1, 1)$, $IS = 2$.

12. $NT = 4$,第 4 棵生成树$[e_2, e_3, e_4]$,如图 6.6(e).

13. $IT = 1$,已得全部生成树:

$$[e_1,e_2,e_3],\ [e_1,e_2,e_4],\ [e_1,e_3,e_4],\ [e_2,e_3,e_4].$$

停.

§6.8 求基本圈

设 G 是连通 (n,m) 图,任取 $v\in V(G)$,则由广度优先搜索法可产生以 v 为根的外向树 T. 由定理2.1,对每条弦 e,$T+e$ 有唯一圈,称为基本圈,这样得到的 $(m-n+1)$ 个基本圈正好构成圈基.

在广度优先搜索中,从 $v\in V(G)$ 开始的搜索,需检验与 v 关联的每一条边. 当边 (u,w) 生长到树中时,给顶点 w 两个标号:$l(w)$ 表在树 T 上从根 v 到 w 的距离,即 T 上唯一的 v-w 路的长;$p(w)$ 表该 v-w 路上 w 的紧前顶点. 不在树 T 上的所有顶点 x,令

$$l(x)=-1.$$

今设 $u\in V(T)$,边 (u,w) 是正在检验的边,有两种可能.

(1) $l(w)=-1$,表示 u 在 T 上,w 不在 T 上,故边 (u,w) 可生长到 T 中.

(2) $l(w)\neq-1$,表示 u 与 w 都在 T 上,故加上边 (u,w) 后有唯一圈. 这时,从顶点 u 与 w 各自利用 p 标号逆向追踪可得唯一的 u-w 路:

$$u,\ p(u),\ p(p(u)),\ \cdots,\ p(w),\ w,$$

于是这个圈便找到了.

算法 6.10 求基本圈的广度优先搜索法

设 G 是连通 (n,m) 图,$V=\{1,2,\cdots,n\}$,H 是邻接矩阵,T 表当前得到的树,开始时 $T=\varnothing$,称顶点 1 为树根,TW 为队列,并令

$$p(i)\triangleq 树\ T\ 上顶点\ i\ 的父顶点,$$

$$TW\triangleq 存在未检验关联边的顶点子集,$$

$$l(i)=\begin{cases}T\ 上根到顶点\ i\ 的距离, & i\in V(T),\\ -1 & ,i\notin V(T),\end{cases}$$

在开始,$TW=\{1\}$,$l(1)=0$,$l(i)=-1$,$2\leqslant i\leqslant n$.

第 1 步 若 $TW=\varnothing$,终止,否则取队头 $i\in TW$.

第 2 步 检验与顶点 i 关联的每一条边. 若 $H_{i\cdot}=0$,则令 $TW=TW\backslash\{i\}$,转第 1 步. 否则取顶点 i 的一条关联边 (i,j).

第 3 步 若 $l(j)=-1$,转第 4 步. 否则输出基本圈:

$$i,\ p(i),\ p(p(i)),\ \cdots,\ p(j),\ j,\ i,$$

转第 5 步.

第 4 步 令 $T=T+(i,j)$, j 入队列 TW,

$$p(j)=i,\ l(j)=l(i)+1.$$

第 5 步 删去边 (i,j),即在 H 中令

$$h_{ij}=h_{ji}=0,$$

转第 2 步.

注 算法 6.10 是以 G 连通为条件的. 在一般情形(即无论 G 是否连通) 可作如下修改:

在第 1 步, 当 $TW = \varnothing$ 时, 不是终止而是问是否有 $x \in V(G), l(x) = -1$. 若有, 继续迭代, 否则终止. 在终止时得到各连通支的基本圈.

例 6.11 求图 6.1(a) 的图 G 的基本圈.

由算法 6.10, 所产生的广度优先生成树如图 6.5.

第 1 个基本圈的产生在虚边(4,5), 分别从端点 4,5 逆向追踪, 他们的第一个公共前代顶点是顶点 2, 故得基本圈

$$[4,2,5].$$

第 2 个基本圈, 分别从虚边(7,8) 的端点 7,8 逆向追踪, 第一个公共前代顶点是顶点 3, 故得基本圈

$$[7,3,6,8].$$

第 3 个基本圈, 分别从虚边(9,10) 的端点 9,10 逆向追踪, 第一个公共前代顶点是顶点 8, 故得基本圈

$$[9,8,10].$$

§6.9 求有向圈

设 G 是(n,m) 有向图. 不妨设 G 连通, 否则可在各连通支进行. 其次, 可设 G 没有出度或入度为零的顶点, 因为这样的顶点不可能在任何有向圈上.

设图 G 的顶点编号为 $1,2,\cdots,n$. 今从顶点 p_1 开始搜索有向路 $P = (p_1, p_2, \cdots, p_k)$, 直到 P 不能再延伸, 这时, 若有边(p_k, p_1), 则得一有向圈$[p_1, p_2, \cdots, p_k]$, 否则, 按深度优先搜索的原则, 退回到顶点 p_{k-1}, 看能否从 p_{k-1} 沿另外的边再把有向路$(p_1, \cdots, p_{k-1})$ 延伸. 由此, 当退回到顶点 p_1 时, 从 p_1 出发的所有有向路都检验了, 并相应得到了有向圈. 然后从下一个顶点开始, 重复上述过程, 直到求出所有有向圈. 在这一过程中应注意下列三点.

1. 为使从 p_1 出发的搜索得到含 p_1 的有向圈(如果存在), 在有向路的延伸中不能到达已经到达过的顶点.

2. 应当避免有 q 个顶点的有向圈产生 q 次——分别从每个顶点出发各得一个有向圈. 设 p_1 是初始顶点, 则只要在有向路的延伸中不出现 $i \leqslant p_1$ 的顶点 i 就可办到. 换言之, 所产生的有向圈其初始顶点的编号是最小的, 这样就能保证有向圈只产生一次.

3. 在有向路的延伸中, 同一条路不能使用多于一次. 为此设计一个 $n \times n$ 禁止矩阵 $X = (x_{ij})$,

$$x_{ij} = \begin{cases} 1, & \text{禁止从顶点 } i \text{ 到 } j, \\ 0, & \text{否则}. \end{cases}$$

在开始时, $X = 0$, 当从 p_k 退回到 p_{k-1} 时, 则禁止从顶点 p_{k-1} 到 p_k, 即令 $x_{p_{k-1}, p_k} = 1$.

上述步骤可在邻接矩阵上实现.

算法 6.11 求有向圈的深度优先搜索法

设 G 是连通(n,m) 有向图, $V = \{1,2,\cdots,n\}$, H 是邻接矩阵, P 是阶 n 的一维数组,

$$P = (p_1, p_2, \cdots, p_k, 0, \cdots, 0),$$

表示在搜索过程中当前有向路，其起点为 p_1，终点为 p_k. 在开始，

$$P = (0, 0, \cdots, 0),\ k = 1,\ p_1 = 1.$$

第 1 步　$X = 0$.

第 2 步　$i = p_1 + 1$.

第 3 步　若 $i > n$，转第 8 步.

第 4 步　若没有边 (p_k, i) 即 $H[p_k, i] = 0$，转第 7 步.

第 5 步　若顶点 i 在 P 上，转第 7 步.

第 6 步　若禁止从顶点 p_k 到 i 即 $X[p_k, i] = 1$，转第 7 步. 否则令

$$k = k + 1,\ p_k = i,$$

转第 2 步.

第 7 步　令 $i = i + 1$，转第 3 步.

第 8 步　若有边 (p_k, p_1) 即 $H[p_k, p_1] = 1$，则输出有向圈

$$[p_1, p_2, \cdots, p_k].$$

第 9 步　若 $k = 1$，令

$$p_1 = p_1 + 1,\ p_k = p_1,$$

转第 11 步.

第 10 步　令

$$X[p_k, j] = 0,\ j = 1, \cdots, n,$$
$$X[p_{k-1}, p_k] = 1,\ p_k = 0,\ k = k - 1,$$

转第 2 步.

第 11 步　若 $p_1 > n$，停.

若 $p_1 = n$，转第 8 步.

若 $p_1 < n$，转第 1 步.

例 6.12　求图 6.7 的有向图的有向圈.

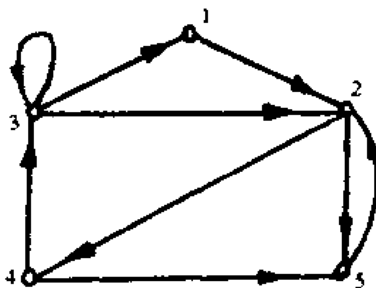


图 6.7

利用算法 6.11，其迭代过程如表 6.3 ～ 表 6.7. 由此得图 6.7 的 5 个有向圈：

$$v_1v_2v_4v_3v_1,\ v_2v_4v_3v_2,\ v_2v_4v_5v_2,\ v_2v_5v_2,\ v_3v_3.$$

表 6.3　从顶点 1 开始搜索

p_1	p_2	p_3	p_4	p_5	记录圈和 x_{ij}
1	0	0	0	0	
1	2	0	0	0	
1	2	4	0	0	
1	2	4	3	0	圈[1,2,4,3],令 $h_{43}=1$
1	2	4	0	0	
1	2	4	5	0	令 $x_{45}=1$
1	2	4	0	0	令 $x_{41}=x_{42}=x_{43}=x_{44}=x_{45}=0,x_{24}=1$
1	2	0	0	0	
1	2	5	0	0	令 $x_{25}=1$
1	2	0	0	0	
1	0	0	0	0	

表 6.4　从顶点 2 开始搜索

p_1	p_2	p_3	p_4	p_5	记录圈和 x_{ij}
2	0	0	0	0	
2	4	0	0	0	
2	4	3	0	0	圈[2,4,3],令 $h_{43}=1$
2	4	0	0	0	
2	4	5	0	0	圈[2,4,5],令 $h_{45}=1$
2	4	0	0	0	令 $x_{41}=x_{42}=x_{43}=x_{44}=x_{46}=0,x_{24}=1$
2	0	0	0	0	
2	5	0	0	0	圈[2,5],令 $x_{25}=1$
2	0	0	0	0	

表 6.5　从顶点 3 开始搜索

p_1	p_2	p_3	p_4	p_5	记录圈和 x_{ij}
3	0	0	0	0	圈[3]

表 6.6　从顶点 4 开始搜索

p_1	p_2	p_3	p_4	p_5	记录圈和 x_{ij}
4	0	0	0	0	
4	5	0	0	0	令 $x_{46}=1$
4	0	0	0	0	

表 6.7 从顶点 5 开始搜索

p_1 p_2 p_3 p_4 p_5	记录圈和 x_{ij}
5 0 0 0 0	

§6.10 可分性算法

若图 G 有环或平行边，则可去掉而不影响可分性. 若图 G 不连通，则可对各连通支分别进行讨论. 若 G 有悬点，除开只有一条边的图，与悬点相邻的顶点必为分离点. 因此，为简单计且不失一般性，可假定图 G 是简单连通图且没有悬点.

可分性算法是利用定理 2.10 的一个结果：图 G 的两条边在同一个块中，当且仅当该两边在同一个圈上.

引理 6.1 图 G 的两个基本圈的非空交是一条路(习题 6.24).

引理 6.2 设 C_1 与 C_2 是两个不同的基本圈且

$$e_1 \in E(C_1)\backslash E(C_2),\ e_2 \in E(C_2)\backslash E(C_1),$$

e_3 是 C_1 与 C_2 的公共边，则必有圈含 e_1 与 e_2(习题 6.25).

算法 6.12 可分性的基本圈标号法

设 (n,m) 图 G 是简单连通图且没有悬点.

第 1 步 $k=1$，产生第一个基本圈 C_1，C_1 的边都标 2.

第 2 步 若 $k=m-n+1$，停. 否则令 $k=k+1$.

第 3 步 产生第 k 个基本圈 C_k，并且

(1) 若 C_k 的边都未标号时，设 $C_1,\cdots,C_{k-1}$ 的边的最大标号是 t，则 C_k 的边都标 $t+1$，转第 2 步.

(2) 若 C_k 的边有标号的，其标号为

$$i_1 < i_2 < \cdots < i_s,$$

则图 G 的所有标号为 $i_1,i_2,\cdots,i_s$ 的边都改标号为 i_1，C_k 的未标号的边也都标 i_1，转第 2 步.

引理 6.3 当算法 6.12 终止时，任二边有相同标号，当且仅当该二边在同一个圈上(不必是基本圈)(习题6.26).

由定理 2.10 和引理 6.3，具有同一标号的边构成图 G 的一个块. 于是最终得到多少个不同标号，图 G 就有多少个块(不包括桥).

引理 6.4 顶点 v 是分离点，当且仅当与 v 关联的边至少有两个不同的标号(习题 6.27).

引理 6.5 没有得到标号的边是桥(习题 6.28).

例 6.13 判定图 6.1(a) 的图 G 的可分性.

由例 6.11，　圈[4,2,5] 的边都标 2；

　　　　　　圈[7,3,6,8] 的边都标 3；

　　　　　　圈[9,8,10] 的边都标 4；

没有标号的边(1,2) 与(1,3) 是图 G 的两座桥. 于是图 G 有 4 个分离点，5 个块，如图 6.8.

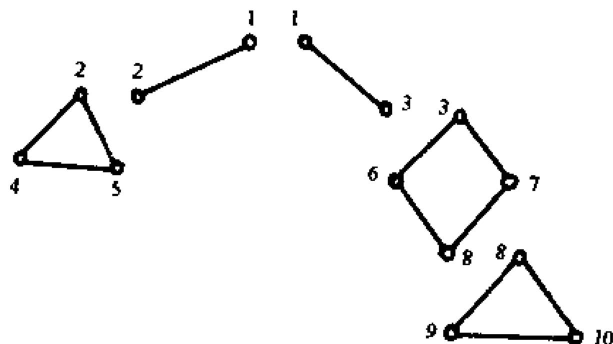


图 6.8

习 题 六

6.1 对图 6.9 的无向图 G，分别用边目录法，二数组法和后继表法存贮.

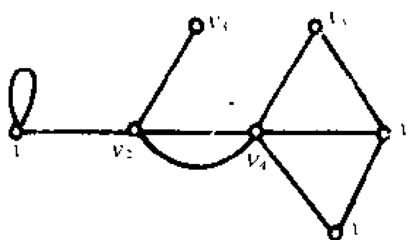


图 6.9

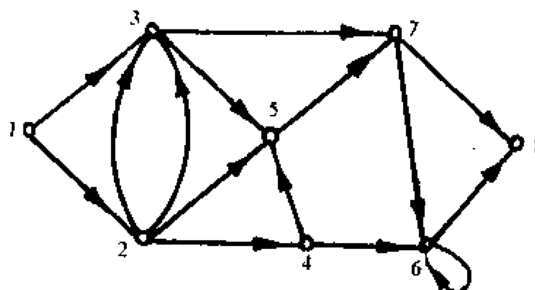


图 6.10

6.2 对图 6.10 的有向图 G，分别用边目录法，二数组法和后继表法存贮.

6.3 若在计算机中存贮了图 G 的关联矩阵，如何转化为邻接矩阵，边目录，二数组与后继表的存贮？试分别就无向图与有向图进行讨论.

6.4 输入图的邻接矩阵，如何转化为关联矩阵，边目录，二数组与后继表的输入？试分别就无向图与有向图进行讨论.

6.5 用二数组输入图 G，如何转化为关联矩阵，邻接矩阵，边目录与后继表输入，分别讨论无向图与有向图的情形.

6.6 图 G 的后继表与边目录如何互相转换，分别讨论有向图与无向图的情形.

6.7 用算法 6.1 遍历图 6.9 的无向图.

6.8 用算法 6.1 遍历图 6.10 的有向图.

6.9 用算法 6.2 遍历图 6.9 的无向图.

6.10 用算法 6.2 遍历图 6.10 的有向图.

6.11 用算法 6.3 求图 G 的连通性，已知图 G 的邻接矩阵是

$$H=\begin{bmatrix}0&0&0&1&0&1&0&0&1\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&1&0\\1&0&0&1&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\1&0&0&1&0&0&0&0&1\\0&0&1&0&1&0&0&1&0\\0&0&1&0&0&0&1&0&0\\1&0&0&0&0&1&0&0&0\end{bmatrix}.$$

6.12　用算法 6.5 求图 6.11 的有向图的强连通性.

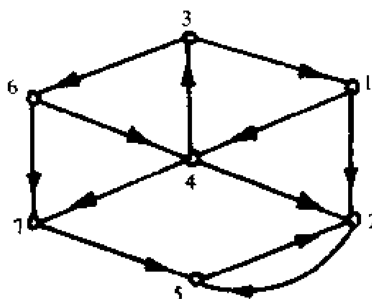


图 6.11

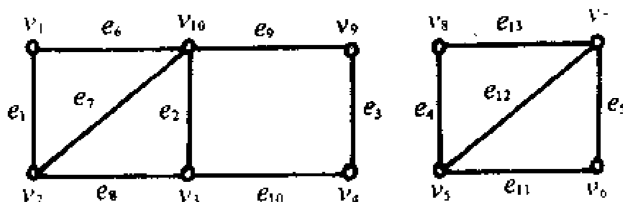


图 6.12

6.13　如图 6.12,用算法 6.6 求生成林.

6.14　设图 G 的后继表如下,试用算法 6.7 求深度优先生成树.

1: 2,6,11	9: 6,7,8,10
2: 1,3,4	10: 6,8,9
3: 2,4,5	11: 1,12,14
4: 2,3,5	12: 11,13,14,15
5: 3,4	13: 12,15
6: 1,7,9,10	14: 11,12,15
7: 6,8,9	15: 12,13,14
8: 7,9,10	

6.15　设图 G 的后继表如习题 6.14,试用算法 6.8 求广度优先生成树.

6.16　用二数组法存贮图 G,e 是 G 的桥,试用二数组法存贮 $G\cdot e$.

6.17　如图 6.13,用算法 6.9 求图的全部生成树.

6.18　对图 6.13 的图 G,写出基本关联矩阵,然后求出基本圈矩阵,再由此写出基本圈.

6.19　对图 6.13 的图 G,由算法 6.10 求基本圈.

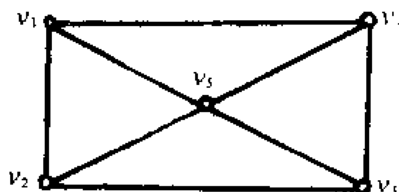


图 6.13

6.20 用算法 6.4 求图 2.14 的有向图的可达矩阵.

6.21 算法 6.4 对无向图是否适用?试求图 6.12 的无向图的可达矩阵.

6.22 给出图的邻接矩阵

$$H=\begin{bmatrix}0&0&0&0&1&0\\1&1&0&0&0&1\\0&1&0&1&0&0\\0&0&0&0&1&0\\0&0&0&1&0&1\\1&0&1&1&0&0\end{bmatrix},$$

试用算法 6.11 求所有有向圈.

6.23 利用深度优先生成树如何求有向圈?

(1) 给出计算步骤;

(2) 由(1)的算法重求习题 6.22 图 G 的有向圈.

6.24 证明引理 6.1.

6.25 证明引理 6.2.

6.26 证明引理 6.3.

6.27 证明引理 6.4.

6.28 证明引理 6.5.

6.29 给定连通图 G 的邻接矩阵

$$H=\begin{bmatrix}0&1&1&1&0&0&0&0&0\\1&0&0&1&0&0&0&0&0\\1&0&0&1&0&0&0&0&0\\1&1&1&0&1&0&0&0&0\\0&0&0&1&0&1&0&0&1\\0&0&0&0&1&0&1&1&1\\0&0&0&0&0&1&0&1&0\\0&0&0&0&0&1&1&0&1\\0&0&0&0&1&1&0&1&0\end{bmatrix},$$

试用算法 6.12 求图的可分性.

6.30 设 T 是连通图 G 的深度优先生成树,则顶点 v 是分离点的充要条件是下列条件之一成立

(1) v 是根且有一个以上子顶点.

(2) v 不是根但存在 v 的子顶点 u,没有从 u 或 u 的后代顶点到 v 的前代顶点的返回边.

6.31 设 T 是连通图 G 的深度优先生成树,每个顶点 i 有一个 l 标号:

$$l(i)\triangleq \text{顶点 } i \text{ 生长到 } T \text{ 的顺序号},$$

令

$$r(v)=\min\{l(v),\{l(w)\mid w\text{ 是 }v\text{ 的前代顶点，且存在从 }v\text{ 或 }v\text{ 的后代顶点到 }w\text{ 的返回边}\}\}.$$

试证顶点 v 是分离点的充要条件是存在 v 的子顶点 u 使

$$l(u)\geqslant r(v).$$

6.32 (1) 对习题 6.14 中所求出的深度优先生成树，由 l 标号求出 r 标号；

(2) 由(1)的结果求图 G 的分离点，桥和块.

6.33 试给出由深度优先搜索树求图的可分性的算法.

6.34 试给出由深度优先搜索树求图的强连通性的算法.

6.35 用习题 6.34 的算法重解例 6.7.

第七章　图论模型

§7.1　欧拉图

在第一章已经指出，用图论方法建立的第一个模型就是七桥问题. 七桥问题的解决是非常完满的.

定义 7.1　包含图 G 的所有边的迹称为**欧拉迹**，包含图 G 的所有边的闭迹称为**欧拉闭迹**. 假定图 G 没有孤立点，若 G 含有欧拉闭迹，则称 G 是**欧拉图**.

定理 7.1　图 G 是欧拉图的充要条件是，G 连通且没有奇顶点.

证　必要性是明显的，今证充分性. 任取 $v_0 \in V(G)$，从 v_0 沿 G 的没有经过的边旅行，由于顶点度是偶数，故途经之顶点能进则能出. 再由有限性，旅行必终止于出发点 v_0，因而得一闭迹. 设 C 是边数最多的闭迹. 若 C 不是欧拉圈，则 $E(G)\backslash E(G) \neq \varnothing$. 由连通性，存在 $u_0 \in V(C)$，

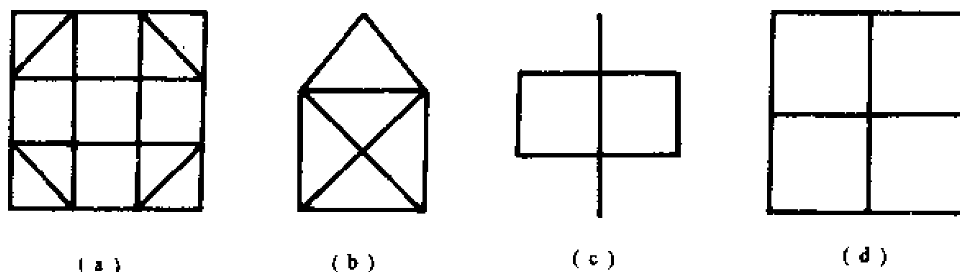
$$d_G(u_0) > d_C(u_0),$$

现从 u_0 出发沿 $G-C$ 的边旅行，同样旅行终止于出发点 u_0 得一闭迹 C'. 今从 v_0 出发沿 C 到达 u_0，再沿 C' 回到 u_0，最后由 C 回到 v_0，由此所得闭迹其边数多于 C 的边数，矛盾.　**证毕.**

从定理 7.1 可导出关于一笔画的有趣结果. 一个图可以一笔画是指从图的某个顶点出发，经每条边恰好一次，最后终止在出发点或另一个顶点，即该图可以一笔画成.

推论 7.2　一个图能一笔画成的充要条件是，G 连通且奇顶点数为 0 或 2(习题 7.1).

例 7.1　图 7.1(a) 连通，没有奇顶点，故是欧拉图. 图 7.1(b) 与(c) 连通，有两个奇顶点，故不是欧拉图，但能一笔画. 图 7.1(d) 有 4 个奇顶点，故不是欧拉图，也不能一笔画.



(a)　(b)　(c)　(d)

图 7.1

定理 7.1 指出，七桥问题的答案是否定的，即图 1.2 的图不是欧拉图. 哥尼斯堡的人要把七座桥恰好经过一次的环城游览是不可能的.

定理 7.1 告诉我们，判断一个图是否欧拉图或者是否可以一笔画都是容易的. 但如何能实现一笔画?注意到定理 7.1 的证明是构造性的，证明的方法——逐步扩大闭迹直至得

到欧拉闭迹——就是实现的步骤. 由此即得如下算法.

算法 7.1　一笔画算法

设 $G=(V,E)$ 是连通 (n,m) 图. 用 a 和 b 分别表示一笔画的起点和终点. 若有二个奇顶点, 则可从 a 到 b 一笔画. 若没有奇顶点, 令 $a=b$, 则可从 a 到 a 一笔画.

以 $(m+1)$ 维有序表 $l(\cdot)$ 表示欧拉迹. $l(u)$ 表示紧接边 u 进入欧拉迹的边. $l(m+1)$ 表示欧拉迹上进入的第 1 条边, $l(u)=-1$ 表示边 u 是最后进入欧拉迹的边.

$r(i)=u$ 表示边 u 是欧拉迹上最后到达顶点 i 的边, $i=1,2,\cdots,n$.

$N(i)$ 表示顶点 i 的关联边集. 若 $u\in N(i)$, 则 $L_i(u)=j$ 表示边 u 的另一端点是顶点 j. $N(i)$ 所含边的数目用 $h(i)$ 表示.

开始　设 $a,b\in V$ 是欧拉迹的端点(对欧拉闭迹, $a=b$).

$$l(i)=0,\ i=1,2,\cdots,m,m+1.$$
$$r(i)=0,\ i=1,2,\cdots,n,\ r(a)=m+1.$$
$$u=m+1,\ w=-1.$$
$$k=0\ (k\ \text{表示进入欧拉迹的边数}).$$
$$i=a,\ t=b.$$

第 1 步　若 $h(i)=0$, 转第 5 步.

第 2 步　设 e 是 $N(i)$ 的第 $h(i)$ 条边, 令

$$h(i)=h(i)-1,$$

若 $e=u$ 或 $l(e)\neq 0$, 转第 1 步.

第 3 步　$j=L_i(e)$ 是边 e 的另一端点, 令

$$r(j)-e,\ l(u)=e,$$
$$k=k+1,\ i=j,$$

若 $i=t$, 转第 4 步, 否则令 $u=e$ 转第 1 步.

第 4 步　令 $l(e)=w$. 若 $k=m$, 由 $l(\cdot)$ 得欧拉迹, 停.

第 5 步　选择顶点 v 使

$$h(v)\neq 0\ \text{且}\ r(v)\neq 0,$$

令

$$i=v,\ t=v,$$
$$u=r(i),\ w=l(u),$$

转第 1 步.

例 7.2　图 7.2 的图 G 其顶点度都是偶数, 故图 G 是欧拉图, 试用算法 7.1 求欧拉闭迹.

$$N(1)=\{e_1,e_2\},$$
$$N(2)=\{e_1,e_3,e_4,e_5\},$$
$$N(3)=\{e_2,e_3,e_6,e_8\},$$
$$N(4)=\{e_4,e_6,e_7,e_{10}\},$$
$$N(5)=\{e_5,e_7,e_{11},e_{12}\},$$

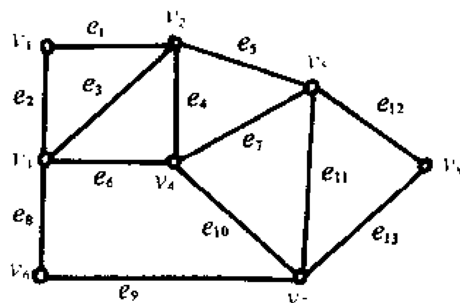


图 7.2

$N(6)=\{e_8,e_9\}$, $N(7)=\{e_9,e_{10},e_{11},e_{13}\}$, $N(8)=\{e_{12},e_{13}\}$.

选 $i=v_1$ 开始，由算法 7.1，得

$$r=(1,3,6,4,7,8,11,13),$$

$$l=(-1,8,1,6,4,3,5,9,13,7,10,11,12,2).$$

于是欧拉闭迹是

$$[e_2,e_8,e_9,e_{13},e_{12},e_{11},e_{10},e_7,e_5,e_4,e_6,e_3,e_1].$$

§7.2 计算机鼓轮设计

例 7.3 计算机旋转鼓轮的位置是借助于鼓轮表面上的一系列电触点所产生的二元信号来识别的. 鼓轮的表面分成 2^n 个扇形. 每个部分由绝缘体或导体组成，绝缘体部分给出信号"0"(没有电流)，导体部分给出信号"1"(有电流). 如图 7.3 中鼓轮的位置由四个触点给出的读数是"0010". 若鼓轮按顺时针方向旋转一个扇形部分，则触点的读数成为"1001". 这两个位置的读数是不同的，因面能够区别开来. 再继续旋转二个扇形部分，其读数是"0010"，这与初始位置的读数相同，故无法区分. 这就产生一个问题，鼓轮上的 2^n 个扇形部分如何安排导体与绝缘体，使相继位于鼓轮周围的 k 个触点能够区分鼓轮的不同位置，并且希望数 k 尽可能地小.

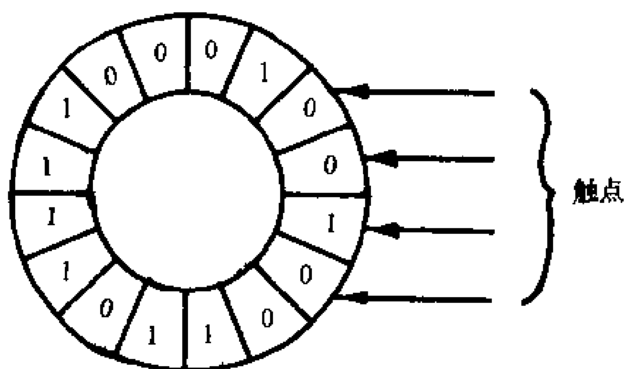


图 7.3 计算机鼓轮($n=4$)

首先注意，k 个触点产生 k 位数字的二进数，一共产生 2^k 个这样的数. 要区别鼓轮的 2^n 个位置，必需使触点所产生的二进数个数满足

$$2^k \geqslant 2^n. \tag{7.1}$$

引理 7.3 区分 2^n 个位置的最小触点数为 n.

证 由式(7.1)，$k\geqslant n$. 下面证明 $k=n$，为此定义一个有向图 D_n：D_n 的顶点是$(n-1)$位数字的二进数 $p_1p_2\cdots p_{n-1}$，其中 $p_i=0$ 或 1. 存在一条边其尾和头分别是

$$p_1p_2\cdots p_{n-1} \text{ 和 } q_1q_2\cdots q_{n-1},$$

当且仅当

$$p_{i+1}=q_i,\ 1\leqslant i\leqslant n-2. \tag{7.2}$$

换言之，D_n 的边都具有形式

$$(p_1p_2\cdots p_{n-1},p_2p_3\cdots p_n), \tag{7.3}$$

并给形如式(7.3)的边以标号 $p_1p_2\cdots p_n$.

(1) D_n 是连通的. 任二顶点 $p_1p_2\cdots p_{n-1}, q_1q_2\cdots q_{n-1}$ 都有边序列

$$(p_1p_2\cdots p_{n-1}, p_2\cdots p_{n-1}q_1)$$
$$(p_2\cdots p_{n-1}q_1, p_3\cdots p_{n-1}q_1q_2)$$
$$\cdots\cdots$$
$$(p_{n-1}q_1\cdots q_{n-2}, q_1q_2\cdots q_{n-1})$$

相连通.

(2) D_n 的每个顶点的出度和入度都是 2. 任一顶点 $p_1p_2\cdots p_{n-1}$ 恰有两个父顶点:

$$0p_1p_2\cdots p_{n-2} \text{ 与 } 1p_1p_2\cdots p_{n-2}.$$

也恰有两个子顶点:

$$p_2p_3\cdots p_{n-1}0 \text{ 与 } p_2p_3\cdots p_{n-1}1.$$

由习题 7.3, D_n 有有向欧拉闭迹. 把这个有向欧拉闭迹看作 D_n 的 2^n 条边的边序列, 边序列标号的第一位数字正好构成一个长为 2^n 的二元序列符合鼓轮表面的设计要求. **证毕.**

算法 7.2 计算机鼓轮设计的算法

第 1 步 作有向图 D_n.

第 2 步 求 D_n 的有向欧拉闭迹.

第 3 步 有向欧拉闭迹的边序列标号的第一位数字产生一个长为 2^n 的二元序列作为鼓轮表面的设计.

例 7.4 当 $n=4$, 由 16 个扇形部分组成的计算机鼓轮, 相应的有向图 D_4 如图 7.4. 在 D_4 中有有向欧拉闭迹

$$[e_1, e_2, e_4, e_8, e_{16}, e_{15}, e_{13}, e_{10}, e_3, e_6, e_{12}, e_7, e_{14}, e_{11}, e_5, e_9],$$

相应边序列标号如表 7.1.

表 7.1

边	标号
e_1	0000
e_2	0001
e_4	0011
e_8	0111
e_{16}	1111
e_{15}	1110
e_{13}	1100
e_{10}	1001
e_3	0010
e_6	0101
e_{12}	1011
e_7	0110
e_{14}	1101
e_{11}	1010
e_5	0100
e_9	1000

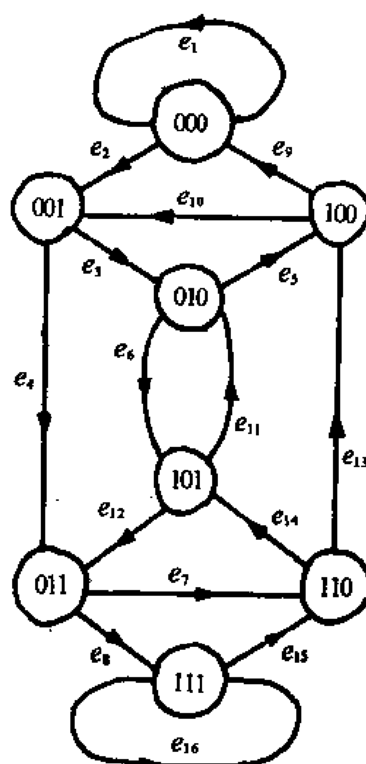


图 7.4 有向图 D_4

从表 7.1 的边序列标号中取第一位数字，得 16 位数字的二元序列

$$0000111100101101, \tag{7.4}$$

按二元序列(7.4) 设计的计算机鼓轮如图 7.5. 其实，由表 7.1，取边序列标号的第二或第三或第四位数字所得二元序列来设计计算机鼓轮也都是一样的.

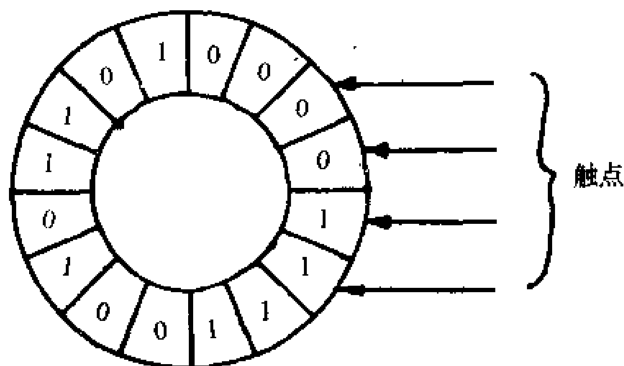


图 7.5 计算机鼓轮设计($n = 4$)

§7.3 道路单行化问题

例 7.5 为改善城市街道交通拥塞，市政部门提出把双向行驶的街道改为单行道. 问题是在改为单行道后能否保证任意两个街口能沿单行道互达？这就是道路单行化问题.

如果把街道口作为顶点，街道作为边，则城市街道系统就对应一个无向图. 改为单行道相当于在这个无向图中给每条边一个方向，由此得到该无向图的定向图. 道路单行化问题用图论的观点来说，就是一个连通图的定向图如何能保证是强连通的？

例 7.6 考查图 7.6(a) 与(b) 的两个道路图. 图 G_1 的任何定向图都不可能是强连通的，因为 G_1 有桥. G_2 的定向图可以是强连通的，如图 7.6(c) 的定向图，因为 G_2 没有桥.

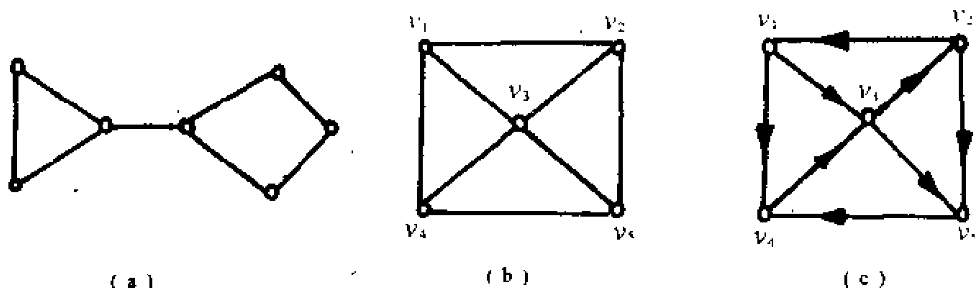


图 7.6 (a) 图 G_1 (b) 图 G_2 (c) G_2 的定向图

例 7.6 具有一般意义，即定理 7.4.

定理 7.4 图 $G = (V,E)$ 有定向图是强连通的，当且仅当 G 连通且没有桥.

证 必要性是明显的，今证充分性. 任取一边 e_0，因为 G 没有桥，由定理 2.7，e_0 在某个圈上. 给此圈定向成为一个有向圈，则定向后的圈是强连通的. 今设 H 是 G 的子图，且 H 的边已定向使 H 成为强连通的. 如果 H 不是 G 的生成子图，则有 G 的顶点不在 H 中. 故由 G 的连通性，存在 G 的边 $e = (x,y)$ 满足

$$e \in E\backslash E(H),\ x \in V(H),\ y \in V\backslash V(H),$$

如图 7.7(a). 同样由定理 2.7 知,边 e 在某个圈 C 上. 今从顶点 x 经边 e 到顶点 y,再继续沿 C 旅行,到达 H 的第一个公共顶点设为 u,如图 7.7(b).

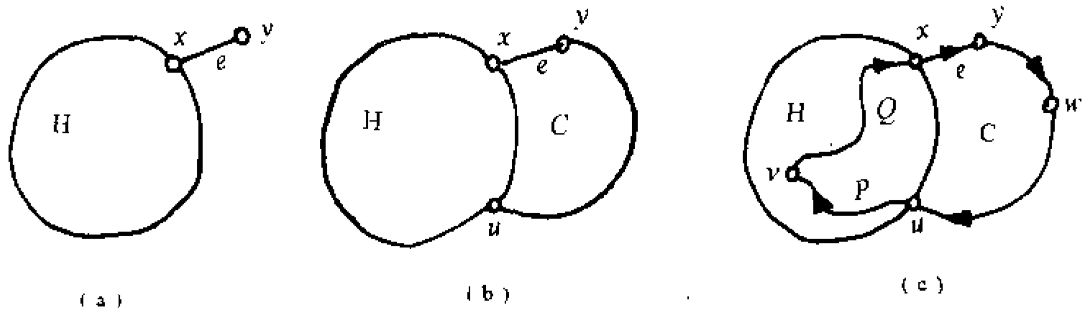


图 7.7

对 C 作如下定向:从 x 沿 e 到 y,再沿 C 到 u,最后回到 x. 对此定向可证 $H \cup C$ 是强连通的. 事实上,对 H 中任一顶点 v,因为 H 强连通,所以在 H 中有 u-v 有向路 P,也有 v-x 有向路 Q. 于是 x 与 C 上任一顶点 w 都能互达,如图 7.7(c).

如果 $H \cup C$ 仍不是 G 的生成子图,可重复上述步骤. 在有限步之后必得 G 的生成子图,其定向图是强连通的. 此时,若图 G 还有边未定向,则可任意定向. 至此,已得 G 的定向图是强连通的. **证毕.**

从定理 7.4 的证明便得到道路单行化的算法.

算法 7.3 道路系统单行化的算法

设道路系统相应的图 G 是连通的.

第 1 步 由可分性算法把图 G 分为块,并标出桥和不是桥的块.

第 2 步 对 G 的桥都用两条方向相反的边代替.

第 3 步 对不是桥的块,按定理 7.4 的证明中的方法,求出生成子图其定向图是强连通的.

第 4 步 若 G 还有边没有定向,则任意定向,终止.

§7.4 储存问题

例 7.7 一家公司制造 n 种化学制品 $A_1, A_2, \cdots, A_n$,其中有些化学制品若放在一起则可能发生危险,如引起爆炸或产生毒气等,称这样的化学制品是不相容的. 为安全起见,在储存这些化学制品时,不相容的不能放在同一储存室内. 问至少要多少个储存室才能存放这 n 种化学制品?

今作图 G, n 种化学制品相应于 n 个顶点 $v_1, v_2, \cdots, v_n$. 顶点 v_i 与 v_j 相邻,当且仅当化学制品 A_i 与 A_j 不相容. 若给图 G 的顶点着色,使任意相邻二顶点有不同的颜色,则化学制品能放在一个储存室的充要条件是相应的顶点着同一种颜色. 换言之,需要多少储存室存放化学制品的问题就是需要多少种颜色给图 G 的顶点着色的问题. 于是储存问题就转化为求图 G 的色数问题.

定理 7.5 设 $G = (V, E)$, $u, v \in V$ 不相邻,则

$$\chi(G)=\min\{\chi(G_1),\chi(G_2)\},\tag{7.5}$$

其中$G_1 \triangleq G+(u,v)$，G_2是G融合顶点u与v的图.

证 设用$\chi(G)$种颜色给G的顶点着色，使G的相邻顶点有不同颜色. 此时若u与v同色，则G_2的相邻顶点有不同颜色，故

$$\chi(G)\geqslant\chi(G_2).\tag{7.6}$$

若u与v不同色，则G_1的相邻顶点有不同颜色，故

$$\chi(G)\geqslant\chi(G_1).\tag{7.7}$$

由式(7.6)与式(7.7)得

$$\chi(G)\geqslant\min\{\chi(G_1),\chi(G_2)\}.\tag{7.8}$$

另一方面由色数的定义直接得到

$$\chi(G)\leqslant\chi(G_1),\tag{7.9}$$

$$\chi(G)\leqslant\chi(G_2),\tag{7.10}$$

由式(7.9)与式(7.10)得

$$\chi(G)\leqslant\min\{\chi(G_1),\chi(G_2)\}.\tag{7.11}$$

由式(7.8)与式(7.11)即得式(7.5). **证毕.**

利用式(7.5)可求图G的色数，即从G得到图G_1与G_2；从G_1又可得$(G_1)_1$与$(G_1)_2$，对G_2也可得$(G_2)_1$与$(G_2)_2$；继续这一过程，直到使所得出的图的任意两个顶点都相邻为止，这种图的色数显然就是图的顶点数.

产生G_1与G_2的过程可以在邻接矩阵上实现.

以上讨论可得色数的算法.

算法 7.4 求色数的深度优先搜索法

设G是(n,m)图，H是邻接矩阵，S是一个栈，存放在各层所生成的G_2，开始为空栈，$k=n$.

第 1 步 在H中若存在$i<j$，$h_{ij}=0$，转第 4 步.

第 2 步 设H的行数是r，若$r<k$，令$k=r$.

第 3 步 若栈空，停，色数$\chi=k$，否则退栈，令H为从栈中退出的H_1，转第 1 步.

第 4 步 生成G_2，即H的第j行加到第i行(逻辑加)，第j列加到第i列(逻辑加)，并在H中去掉第j行第j列，得到的新的邻接矩阵记为H_1并入栈. 记录融合顶点i所包含的顶点.

第 5 步 生成G_1，即在H中令

$$h_{ij}=h_{ji}=1,$$

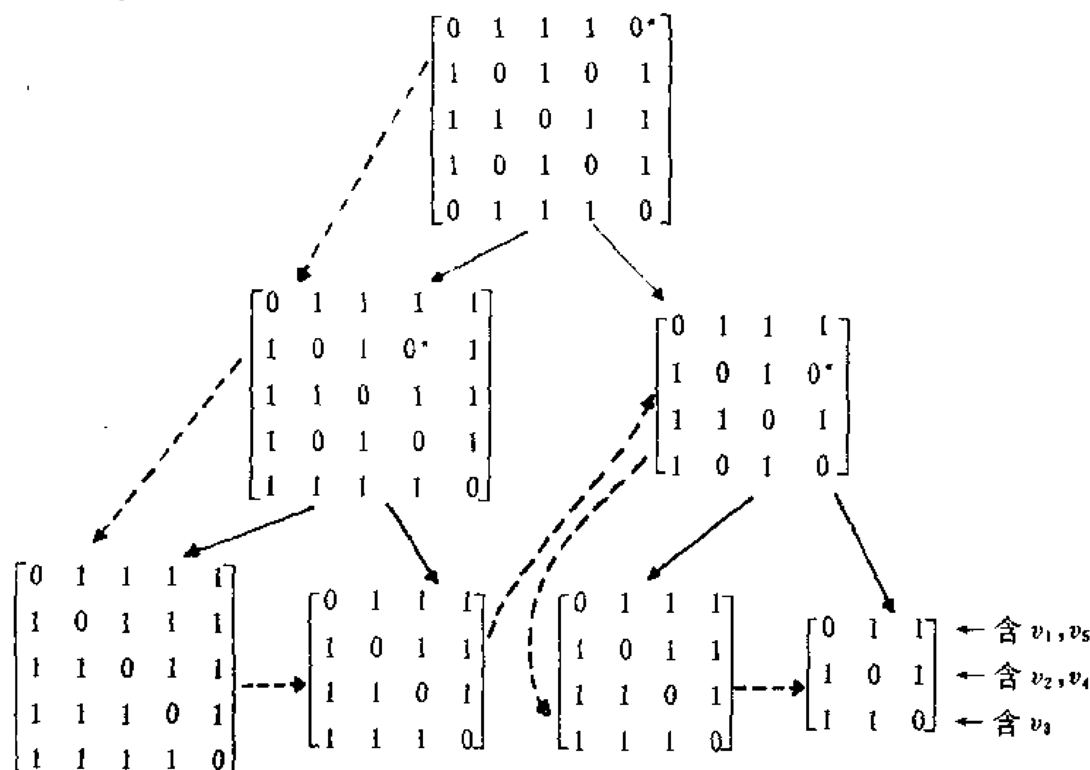
转第 1 步.

例 7.8 求图 7.6(b) 的图的色数.

从图 7.6(b) 的图的邻接矩阵开始，先生成G_2的邻接矩阵，放在右边，入栈；再生成G_1的邻接矩阵，并进行处理. 实线箭头表示这种生成G_2与G_1的过程. 当G_1已处理完毕，即相应的邻接矩阵满足

$$h_{ij}=h_{ji}=1,\ i\neq j$$

的时候退栈. 虚线箭头表示深度优先搜索对图的邻接矩阵处理的顺序. 打"*"的元表示第1步的 h_{ij}.



由此得图 7.6(b) 的色数是 3. 即最少要 3 种颜色给顶点着色使相邻顶点有不同颜色. 着色的方法是,融合顶点所包含的所有顶点着同一颜色,不同的融合顶点着不同的颜色. 在这里,融合顶点 v_1 含 v_1, v_5;融合顶点 v_2 含 v_2, v_4;v_3 不是融合顶点,故分别着 3 种颜色的顶点子集是

$$\{v_1, v_5\},\ \{v_2, v_4\},\ \{v_3\}.$$

下面讨论由逻辑算法求色数的方法.

定义 7.2 设 $G=(V,E)$,若 $S \subseteq V$ 是独立集(独立集的定义见习题 4.26),任意给 S 增加一个顶点都不再是独立集,则称 S 为极大独立集. 若 $K \subseteq V$ 是覆盖,从 K 任意去掉一个顶点就不再是覆盖,则称 K 为极小覆盖.

引理 7.6 设 $G=(V,E)$,则 $S \subseteq V$ 是极大独立集的充要条件是,$V \backslash S$ 是极小覆盖(习题 7.9).

若能找出图 G 所有极小覆盖,则由引理 7.6,就找出了图 G 的所有极大独立集,从中选出顶点数最多的就是最大独立集. 对最大独立集中所有顶点可着同一种颜色,即相应的化学制品可放在同一个储存室. 如何找出所有极小覆盖?我们有下面的引理.

引理 7.7 设 $G=(V,E)$,则 $K \subseteq V$ 是极小覆盖的充要条件是,对每个顶点 $v \in V$,恰有下列两个条件之一成立(习题 7.10):

(1) $v \in K$.

(2) $\Gamma(v) \subseteq K$.

引理 7.7 提出了求极小覆盖的逻辑规则:

$$\text{对每个顶点 } v\text{,选择 } v \text{ 或选择 } v \text{ 的所有相邻顶点.} \tag{7.12}$$

规则(7.12)是一种逻辑运算:

选择 u 或 v 即逻辑加"$u+v$";

选择 u 与 v 即逻辑乘"uv".

逻辑加与逻辑乘具有下列性质:

引理 7.8 逻辑加与逻辑乘满足(习题 7.11):

(1) 结合律 $(u+v)+w=u+(v+w)$, (7.13)

$(uv)w=u(vw)$; (7.13)′

(2) 交换律 $u+v=v+u$, (7.14)

$uv=vu$; (7.14)′

(3) 幂等律 $u+u=u$, (7.15)

$uu=u$; (7.15)′

(4) 吸收律 $u+uv=u$, (7.16)

$u(u+v)=u$; (7.16)′

(5) 分配律 $u+vw=(u+v)(u+w)$, (7.17)

$u(v+w)=uv+uw$; (7.17)′

其中打一撇与不打一撇的公式互为对偶,即不打一撇的公式中的加与打一撇的公式中的乘可以互换.

以上讨论可得求最小覆盖和最大独立集的方法.

算法 7.5 求最小覆盖的逻辑算法

设 $G=(V,E)$ 是连通图.

第 1 步 由规则(7.12)建立逻辑式.

第 2 步 由引理 7.8 的逻辑运算性质化简第 1 步的逻辑式.

第 3 步 由第 2 步直接写出全部极小覆盖.

第 4 步 极小覆盖中顶点数最少的是最小覆盖.

算法 7.6 求最大独立集的逻辑算法

设 $G=(V,E)$ 是连通图.

第 1 步 求出 G 的全部极小覆盖.

第 2 步 由引理 7.6 求出 G 的全部极大独立集.

第 3 步 极大独立集中顶点数最多的是最大独立集.

例 7.9 求图 7.2 的最小覆盖和最大独立集.

首先由规则(7.12)建立逻辑式:

$$\begin{gathered}(v_1+v_2v_3)(v_2+v_1v_3v_4v_5)(v_3+v_1v_2v_4v_6)(v_4+v_2v_3v_5v_7)\\(v_5+v_2v_4v_7v_8)(v_6+v_3v_7)(v_7+v_4v_5v_6v_8)(v_8+v_5v_7),\end{gathered} \tag{7.18}$$

用引理 7.8 的运算性质化简式(7.18)得(习题 7.13):

$$v_1v_2v_4v_5v_6v_7+v_1v_2v_4v_5v_6v_8+v_1v_2v_4v_6v_7v_8+v_2v_3v_4v_5v_6v_8$$

$$+ v_2v_3v_4v_7v_8 + v_2v_3v_5v_7 + v_1v_3v_4v_5v_6v_8 + v_1v_3v_4v_5v_7, \tag{7.19}$$

由式(7.19)直接写出 8 个极小覆盖:

$$K_1 = \{v_1, v_2, v_4, v_5, v_6, v_7\}, \quad K_2 = \{v_1, v_2, v_4, v_5, v_6, v_8\},$$
$$K_3 = \{v_1, v_2, v_4, v_6, v_7, v_8\}, \quad K_4 = \{v_2, v_3, v_4, v_5, v_6, v_8\},$$
$$K_5 = \{v_2, v_3, v_4, v_7, v_8\}, \quad K_6 = \{v_2, v_3, v_5, v_7\},$$
$$K_7 = \{v_1, v_3, v_4, v_5, v_6, v_8\}, \quad K_8 = \{v_1, v_3, v_4, v_5, v_7\},$$

利用引理 7.6 得到 8 个极大独立集:

$$S_1 = \{v_3, v_8\}, \quad S_2 = \{v_3, v_7\},$$
$$S_3 = \{v_3, v_5\}, \quad S_4 = \{v_1, v_7\},$$
$$S_5 = \{v_1, v_5, v_6\}, \quad S_6 = \{v_1, v_4, v_6, v_8\},$$
$$S_7 = \{v_2, v_7\}, \quad S_8 = \{v_2, v_6, v_8\},$$

因此最小覆盖是 K_6,最大独立集是 S_6.

现在回到图的色数问题.

定义 7.3 设图 G 的顶点分解为 $V_1, \cdots, V_k$,每个子集 V_i 中的顶点互不相邻且着同一种颜色,则称 G 有一个 k 着色$(V_1, \cdots, V_k)$. 若 G 的 k 着色满足

(1) V_1 是 G 的极大独立集,

(2) V_i 是 $G - (V_1 \cup V_2 \cup \cdots \cup V_{i-1})$ 的极大独立集,$i = 2, 3, \cdots, k$,

则称这个着色是**典型 k 着色**.

引理 7.9 若图 G 是 k 可着色的,则 G 有典型 k 着色(习题 7.12).

算法 7.7 求色数的逻辑算法

设 $G = (V, E)$ 是连通图.

第 1 步 求 G 的最大独立集 S,S 的顶点着同一种颜色.

第 2 步 若 $G - S \neq \varnothing$,令 $G = G - S$ 转第 1 步.

第 3 步 得 G 的一个典型着色,设着色数目为 k.

第 4 步 若在每步所求最大独立集唯一,则 k 为色数,典型 k 着色的每个子集着同一种颜色. 否则重复 1 至 3 步,求出典型 k 着色的最小的 k,停.

例 7.10 求图 7.2 的色数.

在例 7.9 中已求出 G 的最大独立集

$$S_6 = \{v_1, v_4, v_6, v_8\},$$

令 $V_1 = S_6$,V_1 的顶点着颜色 1.

$G - V_1$ 如图 7.8(a). 先求 $G - V_1$ 的极小覆盖,为此按规则(7.12)建立逻辑式:

$$(v_2 + v_3v_5)(v_3 + v_2)(v_5 + v_2v_7)(v_7 + v_5),$$

化简得 $v_2v_5 + v_2v_7 + v_3v_5$,故 $G - V_1$ 的极小覆盖是

$$K_1 = \{v_2, v_5\}, \ K_2 = \{v_2, v_7\}, \ K_3 = \{v_3, v_5\},$$

相应得出极大独立集

$$S_1 = \{v_3, v_7\}, \ S_2 = \{v_3, v_5\}, \ S_3 = \{v_2, v_7\},$$

他们都是最大独立集.

(1) 令 $V_2 = S_1$，V_2 的顶点着颜色 2. $G-(V_1 \cup V_2)$ 如图 7.8(b)，显然有

$$V_3 = \{v_2\},\ V_4 = \{v_5\},$$

由此得典型 4 着色 (V_1, V_2, V_3, V_4).

(2) 令 $V_2' = S_2$，$G-(V_1 \cup V_2')$ 如图 7.8(c)，显然有

$$V_3' = \{v_2, v_7\},$$

由此得典型 3 着色 (V_1, V_2', V_3').

(3) 令 $V_2'' = S_3$，$G-(V_1 \cup V_2'')$ 如图 7.8(d)，显然有

$$V_3'' = \{v_3, v_5\},$$

由此得典型 3 着色 (V_1, V_2'', V_3'').

比较上面(1)，(2)与(3)，得 G 的色数是 3，分别着不同颜色的顶点子集是

$$\{v_1, v_4, v_6, v_8\},\ \{v_2, v_7\},\ \{v_3, v_5\}.$$

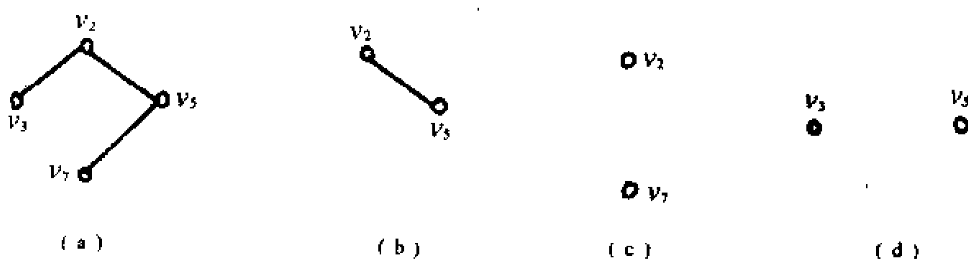


图 7.8　(a) $G-V_1$　(b) $G-(V_1 \cup V_2)$　(c) $G-(V_1 \cup V_2')$　(d) $G-(V_1 \cup V''_2)$

§7.5　排课表问题

例 7.11　在一所学校有 m 位教师 $X_1, \cdots, X_m$ 和 n 个班级 $Y_1, \cdots, Y_n$. 已知教师 X_i 给班级 Y_j 上 w_{ij} 课时，要求制订一张课表使课时尽量少.

设顶点集 $V=(X,Y)$，$X=\{x_1,\cdots,x_m\}$ 对应 m 位教师，$Y=\{y_1,\cdots,y_n\}$ 对应 n 个班级. 顶点 x_i 与 y_j 连接着 w_{ij} 条边，于是得到一个偶图 $G=(X,Y,E)$. 假设在同一个课时，一位教师最多上一个班的课，并且，一个班也最多由一位教师上课，因此在同一个课时的教学时间表对应偶图 G 的一个匹配. 反之，图 G 的每个匹配都对应在一个课时教师上课的一个分派. 换言之，偶图 G 的一个匹配与课表的一个课时正好一一对应. 因此，排课表的问题转化为求偶图的匹配，而使匹配的个数尽量少.

定义 7.4　设 G 是无环图，若能用 k 种颜色给 G 的边着色，使相邻边具有不同颜色，则称 G 是 k **边可着色的**，而这样的边着色称为图 G 的一个**正常 k 边着色**. 图 G 的正常 k 边着色的最小值 k，称为 G **的边色数**，用 $\chi'(G)$ 或 χ' 表示.

对排课表问题现在可如下考虑：每求出 G 的一个匹配，就把匹配中的边着同一种颜色，于是，使匹配的个数尽量少就是使正常边着色的颜色数目尽量少. 换言之，排课表问题就是求相应偶图的边色数的问题.

定理 7.10　设 G 是无环图，则

$$\chi'(G) \geqslant \Delta(G). \tag{7.20}$$

证　因为相邻边具有不同颜色，所以与顶点 v 关联的边必须着不同颜色，故式(7.20)

成立. 证毕.

下面证明,对偶图式(7.20)取等式. 为此先引入如下定义和引理.

定义 7.5 用 k 种颜色任意给图 G 的边着色,称为图 G 的一个 **k 边着色**. 与顶点 v 关联的边若有着颜色 i 的,则称颜色 i 在顶点 v **出现**. 在顶点 v 出现的不同颜色的数目称为顶点 v 的**色度**,记为 $c(v)$. 色度之和 $\sum_{v\in V}c(v)$ 最大的 k 边着色称为**最优 k 边着色**.

引理 7.11 证明(习题 7.18)

(1) 对图 G 的任意 k 边着色,

$$c(v) \leqslant d(v),\ v \in V(G). \tag{7.21}$$

(2) G 有一个正常 k 边着色的充要条件是

$$c(v) = d(v),\ v \in V(G). \tag{7.22}$$

引理 7.12 设 G 连通且不是奇圈,则 G 有一个 2 边着色使

$$c(v) = 2,\ 当\ d(v) \geqslant 2, \tag{7.23}$$

(习题 7.19).

引理 7.13 设$(E_1,\cdots,E_k)$是图 G 的一个最优 k 边着色,若存在 G 的顶点 v 和颜色 i 与 j,使 i 不在 v 出现,j 至少在 v 出现两次,则

$$G[E_i \cup E_j]$$

中含 v 的连通支是奇圈(习题 7.20).

利用所得之引理可以证明,对于偶图,式(7.20)取等号.

定理 7.14 设 G 是偶图,则 $\chi' = \Delta$.

证 设 G 是具有 $\chi' > \Delta$ 的图,$(E_1,\cdots,E_\Delta)$ 是一个最优 Δ 边着色,则存在 $u \in V$, $c(u) < d(u)$. 由引理 7.13,G 含奇圈,与 G 是偶图相矛盾. 证毕.

设 $G = (X,Y,E)$ 是偶图. 如果 $|X| > |Y|$,则在 Y 中增加一些孤立点成为 Y^* 使 $|X| = |Y^*|$,所得偶图为 $G' = (X,Y^*,E)$. 将 X 中的最小度顶点与 Y^* 中的最小度顶点连一条边,重复之,直至得到一个 Δ- 正则偶图 $G^* = (X,Y^*,E^*)$. 由定理 4.7,G^* 有完美匹配. 设 M 是 G^* 的完美匹配,则 $G^* - M$ 是$(\Delta - 1)$- 正则偶图. 用 $G^* - M$ 代替 G^*,重复之,直至求出 G^* 的 Δ 个边不交的完美匹配. 每个完美匹配着一种颜色,得 G^* 的正常 Δ 边着色. 只保留 G 的边,则得 G 的正常 Δ 边着色. 证毕.

算法 7.8 求偶图的边色数的算法

设 $G = (X,Y,E)$是偶图,$|X| > |Y|$,在 Y 中增加$(|X| - |Y|)$个顶点成为 Y^* 得偶图 $G^* = (X,Y^*,E)$.

第 1 步 若 $\Delta(G^*) = \delta(G^*)$,令 $k = 1$ 转第 3 步.

第 2 步 取 $x^0 \in X, y^0 \in Y^*$ 合于

$$d_{G^*}(x^0) = \min\{d_{G^*}(x)\,|\,x \in X\},$$
$$d_{G^*}(y^0) = \min\{d_{G^*}(y)\,|\,y \in Y^*\},$$

令 $G^* = G^* + (x^0,y^0)$,转第 1 步.

第 3 步 任取 G^* 的一个匹配 M.

第 4 步　若 X 的顶点都是 M- 饱和的，则 M 是完美匹配，转第 8 步，否则取 M- 非饱和点 $u \in X$，令

$$S = \{u\}, \ T = \varnothing.$$

第 5 步　取 $y \in \Gamma(S) \backslash T$，若 y 是 M- 非饱和点，转第 7 步.

第 6 步　设 $(y,x) \in M$，则令

$$S = S \cup \{x\}, \ T = T \cup \{y\},$$

转第 5 步.

第 7 步　u-y 路是 M- 增广路，记为 P，令

$$M = M \oplus P,$$

转第 4 步.

第 8 步　若 $k = \Delta$，则得 G^* 的正常 Δ 边着色，只保留 G 的边，得 G 的正常 Δ 边着色，停. 否则令

$$k = k + 1, \ G^* = G^* - M,$$

转第 3 步.

算法 7.8 实际上就是算法 4.1 中增加了从 G 到 G^* 的算法. 从 G 到 G^* 的加边循环次数至多 $|X|\Delta$ 次，而算法 4.1 是多项式算法，故算法 7.8 也是多项式算法.

至此我们完全解决了例 7.11 的排课表问题.

课时尽量少的排课表的方法

(1) 作出排课表问题相应的偶图 G，

(2) 由算法 7.8 求出 G 的正常 Δ 边着色，

(3) 由正常 Δ 边着色得 Δ 课时的课表即为所求.

例 7.11 的问题只有一个要求：课时尽量少. 如果再附加教室的限制呢？

假设总共有 h 节课，安排在一张有 w 课时的课表里，若用 $\{x\}$ 表示不小于 x 的最小整数，则每课时至少要 $\{h/w\}$ 个教室. 下面进一步指出，在一张 w 课时的课表里，总能安排 h 节课，使得在每节课时内最多占用 $\{h/w\}$ 个教室.

引理 7.15　设 M 与 N 是图 G 的两个不相交匹配且 $|M| > |N|$，则存在图 G 的不相交匹配 M' 与 N' 使

$$|M'| = |M| - 1,$$
$$|N'| = |N| + 1,$$
$$M' \cup N' = M \cup N,$$

(习题 7.26).

定理 7.16　设 G 是偶图，$|E(G)| = h, w \geqslant \Delta$，则存在 G 的 w 个不相交匹配 $M_1, \cdots, M_w$ 使

$$E(G) = M_1 \cup \cdots \cup M_w, \tag{7.24}$$

并且对于 $1 \leqslant i \leqslant w$ 有

$$[h/w] \leqslant |M_i| \leqslant \{h/w\}. \tag{7.25}$$

证　由定理 7.14，图 G 的边可划分为 Δ 个互不相交的匹配 $M_1', \cdots, M_\Delta'$. 故对任意 $w \geqslant \Delta$，存在 w 个互不相交的匹配 $M_1', \cdots, M_\Delta', \cdots, M_w'$，其中

$$M_i' = \varnothing,\ i = \Delta + 1,\ \Delta + 2,\ \cdots,\ w,$$

$$E = M_1' \cup M_2' \cup \cdots \cup M_w'.$$

对于这 w 个匹配中边数相差超过 1 的任何两个匹配，反复应用引理 7.15，最后得到图 G 的 w 个匹配满足式(7.24) 与式(7.25)．　证毕．

以上分析可得排课表的一般步骤．

算法 7.9　排课表的算法

设有 m 位教师给 n 个班授课，w_{ij} 表示教师 X_i 给班级 Y_j 授课 w_{ij} 课时，由此得

$$W = (w_{ij})_{m\times n},$$

称为**教学要求矩阵**．教学要求矩阵与偶图

$$G = (X, Y, E)$$

相对应，X 是 m 位教师的顶点，Y 是 n 个班级的顶点，X_i 与 Y_j 间有 w_{ij} 条边．用 h 表示总课时数，则 $h = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}$．

第 1 步　由算法 7.8 求图 G 的正常 Δ 边着色，所得 Δ 个匹配 $M_1, M_2, \cdots, M_\Delta$ 对应一张课时最少，即 Δ 课时的课表．

第 2 步　在第 1 步中所排课表需用教室间数 $\geqslant \{h/\Delta\}$，如有必要，利用引理 7.15，可调整到需用教室间数正好等于 $\{h/\Delta\}$．

第 3 步　如果限制教室间数少于 $\{h/\Delta\}$，则由定理 7.16，应增加课表的课时数．设教室间数 $k < \{h/\Delta\}$，则所需课时数 w 是满足

$$\frac{h}{w} \leqslant k < \frac{h}{w} + 1$$

的最小正整数，即 w 是整数且满足

$$\{h/w\} = k,\ \{h/(w-1)\} > k, \tag{7.26}$$

由此即可排出 w 课时 k 间教室的课表．其方法是增加空匹配 $M_{\Delta+1} = M_{\Delta+2} = \cdots = M_w = \varnothing$，对 w 个匹配 $M_1, M_2, \cdots, M_\Delta, M_{\Delta+1}, \cdots, M_w$，利用引理 7.15，调整到满足式(7.24) 与式(7.25)，从而排出 w 课时 k 间教室的课表。

例 7.12　假设有 4 位教师给 5 个班级授课，其教学要求矩阵是

$$W = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} Y_1 & Y_2 & Y_3 & Y_4 & Y_5 \end{array} \\ \begin{pmatrix} 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

作出相应偶图 G，如图 7.9(a)．增加顶点 x_5 和相应的边得 4- 正则偶图 G^*，如图 7.9(b)．

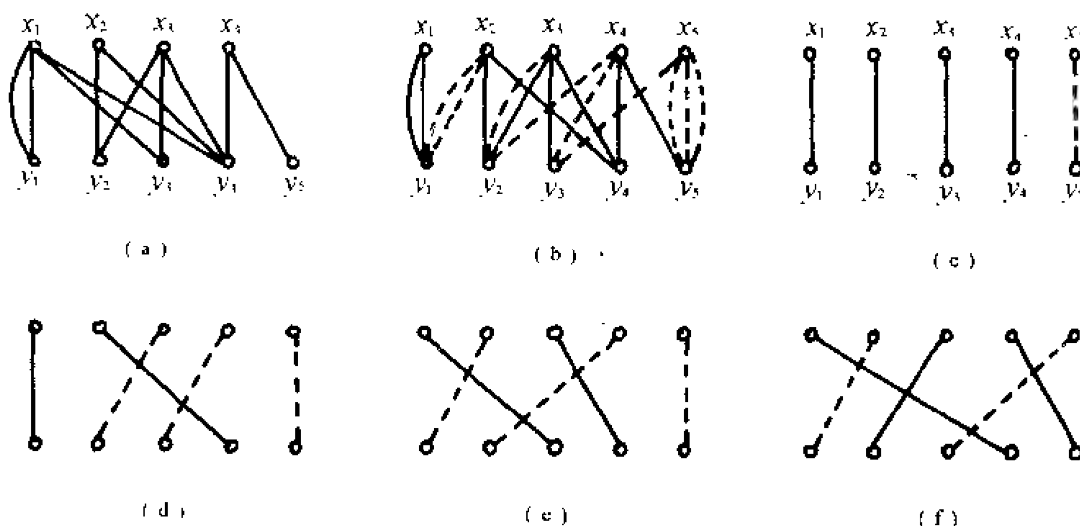


图 7.9 (a) 偶图 G (b) 偶图 G^*，增加的边用虚线表示
(c) 匹配 M_1 (d) 匹配 M_2 (e) 匹配 M_3 (f) 匹配 M_4

用算法 7.8 求出正常 4 边着色，如图 7.9(c) ~ (f). 由此排出 4 课时的课表如表 7.2.

在课表中第 1 节课有 4 个班上课，故需 4 间教室. 因为 $h = 11$，$\Delta = 4$，$[\frac{11}{4}] = 2$，$\{\frac{11}{4}\} = 3$，所以由定理 7.16，能够安排一张课表，授课 4 课时，每一课时有 2 个或 3 个班级在上课，即需要 3 间教室. 其实现步骤是由定理 7.16，对匹配边的数相差大于 1 的两个匹配 M_1 与 M_2，由引理 7.15，作 $G[M_1 \cup M_2]$，如图 7.10(a) 得交错路 $P = y_2x_2y_4x_4$，由对称差得新的匹配 M_1' 与 M_2' 如图 7.10(b).

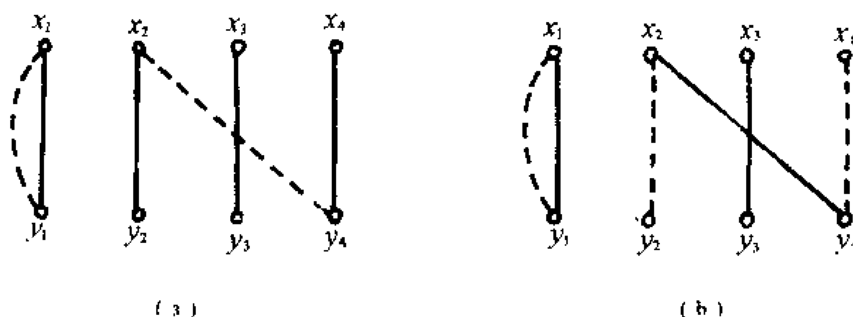


图 7.10 (a) $G[M_1 \cup M_2]$，M_1 的边是实边，M_2 的边是虚边 (b) $G[M_1' \cup M_2']$

由图 7.10(b)，改进表 7.2，得只需 3 间教室的课表如表 7.3.

表 7.2 课表(需 4 间教室)

教师 \ 课时	1	2	3	4
X_1	Y_1	Y_1	Y_3	Y_4
X_2	Y_2	Y_4	—	—
X_3	Y_3	—	Y_4	Y_2
X_4	Y_4	—	—	Y_5

表 7.3 课表(需 3 间教室)

教师 \ 课时	1	2	3	4
X_1	Y_1	Y_1	Y_3	Y_4
X_2	Y_4	Y_2	—	—
X_3	Y_3	—	Y_4	Y_2
X_4	—	Y_4	—	Y_5

表 7.4 课表(需 2 间教室)

教师 \ 课时	1	2	3	4	5	6
X_1	Y_1	Y_1	Y_3	Y_4	—	—
X_2	Y_4	Y_2	—	—	—	—
X_3	—	—	Y_4	Y_2	Y_3	—
X_4	—	—	—	—	Y_4	Y_5

如果只有2间教室，即$k=2$，则满足式(7.26)的$w=6$，所以课表应从4学时增加到6学时. 此时由定理7.16，增加两个空匹配M_5, M_6，反复利用引理7.15得6个新匹配，其所含边数相差不超过1：

$$\{(x_1,y_1),(x_2,y_4)\},\ \{(x_1,y_1),(x_2,y_2)\},$$
$$\{(x_1,y_3),(x_3,y_4)\},\ \{(x_1,y_4),(x_3,y_2)\},$$
$$\{(x_3,y_3),(x_4,y_4)\},\ \{(x_4,y_5)\},$$

相应课表如表7.4.

§7.6 中国邮递员问题

例7.13 假定邮递员从邮局出发经过所投递范围内的每条街道，在递送完邮件之后又返回邮局，问邮递员如何选择投递路线使经过的总的路程最短?这个问题称为**中国邮递员问题**.

如果把投递点作为顶点，所经过的街道作为边，两顶点间的投递距离作为相应边的权，则得到一个非负权的连通图. 于是中国邮递员问题转化为在一个非负权连通图G中求包含G的所有边的权最小的闭通道. 若图G是欧拉图，则欧拉闭迹为所求. 若图G不是欧拉图，则在闭通道中必有某些边是重复的. 设$e\in E(G)$是一条重复的边，则附加一条e的平行边e'，e'的权与e的权相等，我们称e'为e的**复制边**. 因此，当图G不是欧拉图时，就是要增加某些复制边后成为欧拉图，并使复制边的权的和最小. 这里有两个问题：

(1) 如何增加复制边使其成为欧拉图，

(2) 如何使复制边的权和最小.

为此先引入两个引理.

引理7.17 设$G=(V,E)$连通，不是欧拉图，E'是复制边集使

$$G'=(V,E\cup E')$$

是欧拉图. 若v_0是G的奇顶点，则存在G的另一个奇顶点u_0，有一条复制边组成的v_0-u_0路(习题7.28).

引理7.18 设E'是最小权复制边集，G的两个奇顶点v_0与u_0间的由复制边组成的v_0-u_0路是

$$P'=[e_1',e_2',\cdots,e_k'],$$

则

$$P=[e_1,e_2,\cdots,e_k]$$

是G的最短v_0-u_0路(习题7.29).

定理7.19 设$X\subseteq V$是图$G=(V,E)$的奇顶点集，$K(X)$是以X为顶点集的完全图，$K(X)$的每条边(i,j)的权w_{ij}等于在G中最短i-j路的长，则中国邮递员问题的解等价于$K(X)$的最小权完美匹配.

证 中国邮递员问题的解等价于找复制边集E'，使$G'=(V,E\cup E')$是欧拉图，且E'的权和最小. 由引理7.17～7.18，X的顶点两两配对，且$K(X)$的边(i,j)对应G中最短i-j路，因此构成$K(X)$的完美匹配M. 于是E'的权和最小等价于M的权和最小. **证毕.**

由此中国邮递员问题得以解决.

算法 7.10 中国邮递员问题的算法

设中国邮递员问题的模拟图 $G=(V,E)$ 是非负权连通图，所有奇顶点的集是 X.

第 1 步 若 $X=\varnothing$，转第 6 步.

第 2 步 求出 X 的任意两顶点间的距离和最短路.

第 3 步 作出赋权完全图 $K(X)$.

第 4 步 求 $K(X)$ 的最小权完美匹配 M.

第 5 步 对每条边 $(i,j)\in M$，在 G 中复制最短 i-j 路的边，使 G 成为欧拉图 G'，令 $G=G'$.

第 6 步 在欧拉图 G 中求欧拉闭迹即得中国邮递员问题的解.

中国邮递员问题的解可以在图上进行，称为**奇偶点图上作业法**，参看习题 7.53～7.55. 这一方法仅仅对于图的圈较少时是适用的.

§7.7 循环赛排名问题

例 7.14 设有 n 个队参加的循环赛，每次比赛不允许平局，只计胜负，当循环赛结束时如何排名?

今作有向图 G，顶点 i 相应于参赛的第 i 队，若第 i 队胜第 j 队，则作有向边 (i,j)，于是循环赛的结果相应于一个**竞赛图**——基础图是完全图的有向图. 例如，图 7.11 就是有 6 个队参加的循环赛在结束后的一个竞赛图.

由习题 1.42，每个竞赛图都有包含所有顶点的有向路——称为**有向哈密顿路**. 可以设想由有向哈密顿路的顶点顺序来排相应参赛队的名次. 如图 7.11 中有有向哈密顿路 $v_1v_2v_5v_6v_3v_4$. 于是参赛队由此得到的名次依次是一队，二队，五队，六队，三队和四队. 一般地如何在竞赛图中求有向哈密顿路?从下面的引理可得算法.

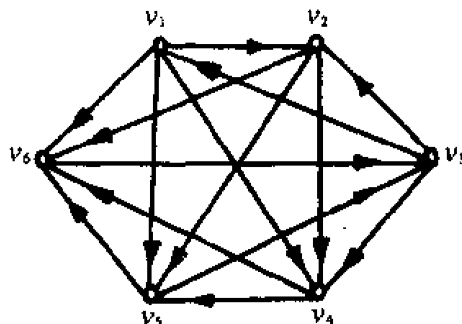


图 7.11 竞赛图 G

引理 7.20 设 G 是竞赛图，$P=v_1v_2\cdots v_k$ 是一条极大有向路(即 P 不能从两端再延伸). 若 P 不是有向哈密顿路，则对任意顶点 $v\notin V(P)$，必有顶点 $v_i(1\leqslant i\leqslant k-1)$ 使

$$(v_i,v),\ (v,v_{i+1})\in E(G),$$

(习题 7.31).

利用这个引理可求竞赛图的有向哈密顿路.

算法 7.11 求有向哈密顿路的算法

设 $G=(V,E)$ 是 n 个顶点的竞赛图.

任取 $u_0\in V$，令 $S=\{u_0\}$，$i=0$.

第 1 步 若 $\Gamma^+(u_i)\cap(V\backslash S)=\varnothing$，令 $j=0$，转第 4 步.

第 2 步　取 $u_{i+1} \in \Gamma^+(u_i) \cap (V \backslash S)$，令

$$S = S \cup \{u_{i+1}\}.$$

第 3 步　若 $i = n - 1$，则

$$P = u_0 u_1 \cdots u_i$$

为有向哈密顿路，停. 否则令 $i = i + 1$，转第 1 步.

第 4 步　若 $\Gamma^-(u_{-j}) \cap (V \backslash S) = \varnothing$，转第 7 步.

第 5 步　取 $u_{-(j+1)} \in \Gamma^-(u_{-j}) \cap (V \backslash S)$，令

$$S = S \cup \{u_{-(j+1)}\}.$$

第 6 步　若 $i + j = n - 1$，，则

$$P = u_{-j} u_{-(j-1)} \cdots u_{-1} u_0 u_1 \cdots u_i$$

为有向哈密顿路，停. 否则令 $j = j + 1$ 转第 4 步.

第 7 步　取 $v \in V \backslash S$，令 $S = S \cup \{v\}$，$k = i$.

第 8 步　若$(u_{k-1}, v) \in E$，则令

$$u_{p+1} = u_p,\ p = i, i-1, \cdots, k+1, k,$$
$$u_k = v,$$
$$i = i + 1,$$

转第 10 步.

第 9 步　令 $k = k - 1$，转第 8 步.

第 10 步　若 $i + j = n - 1$，则

$$P = u_{-j} u_{-(j-1)} \cdots u_{-1} u_0 u_1 \cdots u_i$$

是有向哈密顿路，停. 否则转第 7 步.

例 7.15　由算法 7.11 求图 7.11 的图 G 的有向哈密顿路.

开始　$u_0 = v_3$,　$i = 0$,

2,3　$u_1 = v_4$,　$i = 1$,

2,3　$u_2 = v_6$,　$i = 2$,

1　　$j = 0$,

5,6　$u_{-1} = v_5$,　$j = 1$,

5,6　$u_{-2} = v_1$,　$j = 2$,

7　$v = v_2$,　$k = 2$,

9　　$k = 1$,

8　$u_3 = v_6$,　$u_2 = v_4$, $u_1 = v_2$, $i = 3$,

10　$P = u_{-2} u_{-1} u_0 u_1 u_2 u_3 = v_1 v_5 v_3 v_2 v_4 v_6$ 为所求有向哈密顿路.

由例 7.15 的结果，又可得参赛队的另一排名：一队，五队，三队，二队，四队与六队.

因为竞赛图的有向哈密顿路不一定唯一，所以按此排名是不行的. 不仅如此，我们从习题 1.45 还得知，在竞赛图中有有向哈密顿圈的充要条件是没有孤立集. 竞赛图 7.11 正好没有孤立集且有有向哈密顿圈

$$C = [v_1, v_2, v_4, v_5, v_6, v_3],$$

这就使得此竞赛图的排名可以任意而根本无法排名.

$$H=\begin{bmatrix}0&1&0&1&1&1\\0&0&0&1&1&1\\1&1&0&1&0&0\\0&0&0&0&1&1\\0&0&1&0&0&1\\0&0&1&0&0&0\end{bmatrix},$$

绝对值最大的特征值是 $r=2.232$. H 对应 r 的正特征向量归一化后为

$$\alpha=(0.238,\ 0.164,\ 0.231,\ 0.113,\ 0.150,\ 0.104)^T,$$

于是六个队的排名是：一队，三队，二队，五队，四队与六队.

以上讨论完全解决了顶点大于等于 4 的强连通竞赛图的排名问题. 对于 3 个顶点的强连通竞赛图，3 个队的排名并列.

对于不是强连通图的竞赛图 G，可如习题 1.46 构造凝聚图 G^*，并且容易证明：

定理 7.23 不是强连通图的竞赛图，其凝聚图有唯一有向哈密顿路(习题 7.36).

至此，循环赛排名问题完全解决.

算法 7.12 循环赛排名的算法

设 G 是竞赛图.

第 1 步 求 G 的强连通支，设为 $G_1,\cdots,G_k$，并相应作出凝聚图 G^*，G_i 对应 G^* 的顶点 v_i^*.

第 2 步 对 $i=1,2,\cdots,k$，

当 $|V(G_i)|=3$，G_i 中 3 个顶点相应的队并列，

当 $|V(G_i)|\geqslant 4$，设 G_i 的邻接矩阵是 H_i，求出 H_i 的绝对值最大的特征值的特征向量并归一化，然后按分量从大到小排出相应参赛队名次.

第 3 步 求出 G^* 的唯一有向哈密顿路，排出顶点 $v_1^*,\cdots,v_k^*$ 的顺序.

第 4 步 把第 2 步的排名加入第 3 步的排名序列，得出所有参赛队的排名.

§7.8 旅行推销员问题

在 §6.1 中已经指出，旅行推销员问题是一个还没有解决的难题. 一种求近似解的方法是从一个哈密顿圈出发，然后进行修改，以求得权更小的哈密顿圈.

设 $G=(V,E)$ 是完全图，$|V(G)|=n$，每条边 (v_i,v_j) 有非负权 w_{ij}，$C=v_1v_2\cdots v_nv_1$ 是一个哈密顿圈，则对任意的 $i,j,1<i+1<j<n$ 都可在 C 中去掉边 (v_i,v_{i+1}) 与 (v_j,v_{j+1}) 而附加边 (v_i,v_j) 与 (v_{i+1},v_{j+1}) 得到另一个哈密顿圈 C_{ij}，如图 7.12，

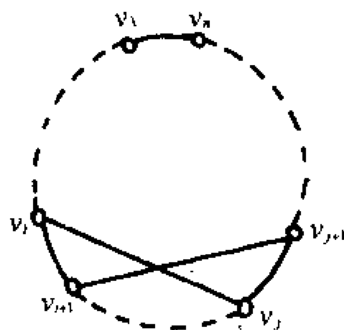


图 7.12

$$C_{ij}=v_1\cdots v_iv_jv_{j-1}\cdots v_{i+1}v_{j+1}v_{j+2}\cdots v_nv_1,$$

因此，若能找到 i,j 满足

$$1 < i+1 < j < n,$$
$$w_{ij} + w_{i+1,j+1} < w_{i,i+1} + w_{j,j+1}, \tag{7.31}$$

则 C_{ij} 的权小于 C 的权.

算法 7.13 旅行推销员问题的近似算法

设 G 是非负权完全图.

第 1 步 找一个哈密顿圈 C.

第 2 步 由式(7.31)找权更小的圈 C_{ij}，转第 3 步. 若不存在这样的圈，转第 4 步.

第 3 步 令 $C = C_{ij}$ 转第 2 步.

第 4 步 终止，或另外选择一个哈密顿圈 C 转第 2 步.

这个算法的近似程度可说明如下：

设 C 是最小权哈密顿圈，则对每个 $v \in V(G)$，$C - v$ 是 $G - v$ 的哈密顿路，自然也是 $G - v$ 的生成树. 设 T 是 $G - v$ 的最小生成树，若 e,f 是 v 的 2 条关联边且使 $w(e) + w(f)$ 尽可能小，则由

$$w(T) \leqslant w(C - v)$$
$$w(C - v) + w(e) + w(f) \leqslant w(C)$$

得 $w(C)$ 的下界

$$w(T) + w(e) + w(f) \leqslant w(C).$$

下面讨论旅行推销员问题的分枝定界法.

注意到完全图 G 的哈密顿圈 C 的等价条件：

(1) C 是含图 G 的所有顶点的连通子图；

(2) $d_C(v) = 2,\ v \in V(G)$.

设想从图 G 的权最小的 n 条边开始，其权和显然是最小权哈密顿圈的一个下界. 如果这 n 条边正好构成哈密顿圈，此即为所求最小权哈密顿圈. 否则进行分枝定界. 先看一个例子.

例 7.18 设旅行推销员问题的权矩阵是

$$W = \begin{bmatrix} 0 & 15 & 5 & 6 \\ 15 & 0 & 11 & 8 \\ 5 & 11 & 0 & 10 \\ 6 & 8 & 10 & 0 \end{bmatrix},$$

试求最小权哈密顿圈.

用二数组 S_1 与 S_2 输入边，用数组 S_3 输入边的权，并把边按权由小到大排列，有

$$S_1 = (1,1,2,3,2,1)$$
$$S_2 = (3,4,4,4,3,2)$$
$$S_3 = (5,6,8,10,11,15)$$

以 C 记最小权哈密顿圈，并令

$$x_{ij} = \begin{cases} 1, & \text{边}(v_i,v_j)\text{在 } C \text{ 上}, \\ 0, & \text{否则}. \end{cases}$$

(1) 取权最小的 4 条边,即令

$$x_{13} = x_{14} = x_{24} = x_{34} = 1,$$

其权和为 29,它是最小权哈密顿圈的一个下界. 由于 $d_C(v_4) = 3$,不合理. 分为两枝:保留 (v_1,v_4),即令 $x_{14} = 1$,或排除 (v_1,v_4),即令 $x_{14} = 0$.

(2) 排除 (v_1,v_4),取权最小的 4 条边,即令

$$x_{13} = x_{24} = x_{34} = x_{23} = 1,$$

其权和为 34. 由于 $d_C(v_3) = 3$,不合理.

(3) 保留边 (v_1,v_4) 时下界为 29;排除边 (v_1,v_4) 时下界为 34;故继续考查保留边 (v_1,v_4) 的情形. 此时排除边 (v_2,v_4) 的权最小的 4 条边是

$$x_{13} = x_{14} = x_{34} = x_{23} = 1,$$

其权和为 32. 因为 $d_C(v_3) = 3$,不合理.

(4) 继续考查保留边 (v_1,v_4) 与 (v_2,v_4) 的情形,此时必须排除边 $(v_3,v$ 、最小的 4 条边是

$$x_{13} = x_{14} = x_{24} = x_{23} = 1,$$

其权和为 30. 这里每个顶点刚好出现 2 次,经检验这正是一个哈密 . 由于在(2) 的下界是 34,在(3) 的下界是 32,故没有继续考查的必要,即进行剪枝. 已求得旅行推销员问题的解:

$$C = v_1v_3v_2v_4v_1,$$

其权为 30.

上述过程可用搜索树表示,如图 7.13.

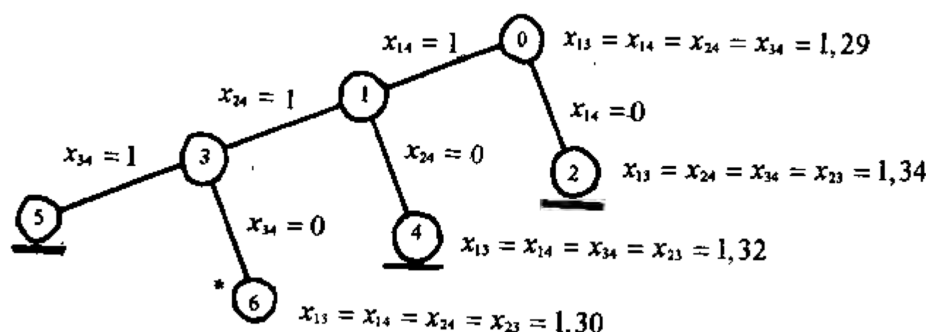


图 7.13　旅行推销员问题的分枝定界过程,打"—"表示剪枝,打"*"表示解

算法 7.14　旅行推销员问题的分枝定界法

设 G 是非负权完全图,$|V(G)| = n$. 用树生长过程产生搜索树.

第 1 步　开始:在顶点 0,选择权最小的 n 条边进入 C,令下界 $a_0 = w(C)$,上界 $b_0 = \infty$.

第 2 步　分枝:选择最小下界的活顶点(活顶点是未查清也未划分的顶点)进行划分,即对 $d_C(v) > 2$ 的顶点 v 的关联边取其权最大的,设为 e,划分为把 e 保留在 C 或排除出 C 两个新顶点从而生长搜索树. 保留在 C 的边称为保留边,并令

$$x_{ij} = \begin{cases} 1, & (i,j) \text{ 是保留边}, \\ 0, & \text{否则}. \end{cases}$$

当排除 v 的关联边 e 时，设 e' 是 $G-C$ 的权最小的边，则令

$$C=(C-e)+e',$$

该顶点的下界为 $w(C)$，转第 3 步. 若没有活顶点，转第 5 步.

第 3 步　查清：若在搜索树上顶点 j 的 C 有顶点与多于 2 条保留边关联，则顶点 j 已查清. 若在搜索树上顶点 j 的 C 是一个哈密顿圈，下界是 a_j，则顶点 j 已查清，并令

$$b_0=\min\{b_0,a_j\}.$$

第 4 步　剪枝：凡满足 $a_j\geqslant b_0$ 的顶点都作为已查清顶点. 转第 2 步.

第 5 步　终止：产生 b_0 的那个顶点为最小权哈密顿圈的顶点.

§7.9　拼花图案

例 7.19　观察大厅、居室或路面的地面装饰，常见一些拼花图案，它们是由同一种正多边形小块拼成的. 问怎样的正多边形小块才能拼成拼花图案?

自然地，我们可以把拼花图案作为一个平面图 G，则 G 有如下性质.

(1) 设 G 有 n 个顶点 m 条边 f 个面，则由 Euler 公式(5.2)，

$$n-m+f=2. \tag{7.32}$$

(2) 每个正多边形小块都是 G 的内部面，可设内部面的边界有 r 条边. 除去无界面边界上的顶点，设有 b 个，剩下的 $(n-b)$ 个顶点的度相同，设为 d，可得

$$d(n-b)<\sum_{v\in V(G)}d(v)=2m<dn,$$

同除以 $2n$，有

$$\frac{d}{2}-\frac{d}{2}\cdot\frac{b}{n}<\frac{m}{n}<\frac{d}{2}. \tag{7.33}$$

(3) 无界面的边界是一个圈，故含有 b 条边，于是由式(5.1)，

$$2m=(f-1)r+b,$$

同除以 nr，整理得

$$\frac{f}{n}=\frac{2}{r}\cdot\frac{m}{n}+\frac{1}{n}-\frac{b}{n}. \tag{7.34}$$

当拼花图案无限扩大时，即当 $n\to\infty$ 时，有

$$\lim_{n\to\infty}\frac{b}{n}=0. \tag{7.35}$$

对式(7.33) 令 $n\to\infty$ 并由式(7.35)，

$$\lim_{n\to\infty}\frac{m}{n}=\frac{d}{2}. \tag{7.36}$$

对式(7.34) 令 $n\to\infty$ 并由式(7.35) 与(7.36)，

$$\lim_{n\to\infty}\frac{f}{n}=\frac{d}{r}. \tag{7.37}$$

今改写式(7.32) 为

$$1+\frac{f}{n}=\frac{m}{n}+\frac{2}{n},$$

令 $n\to\infty$ 并由式(7.36) 与(7.37)，

$$1+\frac{d}{r}=\frac{d}{2},$$

即

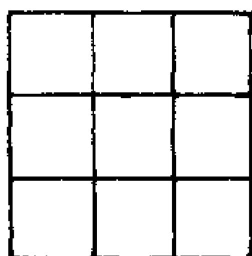
$$(d-2)(r-2)=4. \tag{7.38}$$

满足式(7.38)的正整数 d,r 如表 7.5.

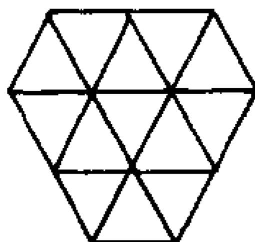
由表 7.5，拼花图案的每一个正多边形小块只能是正 3 边形，正 4 边形与正 6 边形，如图 7.14.

表 7.5

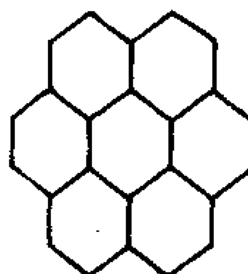
r	3	4	6
d	6	4	3



(a)



(b)



(c)

图 7.14 拼花图案 (a) $r=4, d=4$ (b) $r=3, d=6$ (c) $r=6, d=3$

§7.10 系统监控问题之一

例 7.20 在城市街道中需在街道口设立岗亭，以便监控和指挥过往交通. 为监控所有街道，问至少要设多少岗亭？

用一个连通图来模拟城市街道系统，在某街口设岗亭，即把相应的顶点作为监控点而监控所关联的所有的边，因此，例 7.20 的问题就是一个"顶点监控边"的问题. 换言之，需在一个连通图 G 中找顶点子集 $K \subseteq V(G)$，使每条边都与 K 的一个顶点关联，且 K 是满足这个性质的顶点最少的，也就是求连通图 G 的最小覆盖. 在算法 7.5 中已经提出了求最小覆盖的逻辑算法，下面讨论求最小覆盖的启发式算法.

设想把顶点度越大的顶点作为监控点则越能监控更多的边，这是一个合理的想法.

算法 7.15 最小覆盖的启发式算法 1

设 G 是连通图，用 K 表示所求最小覆盖. 令 $K=\varnothing$.

第 1 步 若 $E(G)=\varnothing$，停. 否则取 $u \in V(G)$ 使

$$d(u)=\max\{d(v) \mid v \in V(G)\}.$$

第 2 步 令 $K=K \cup \{u\}$,

$$G=G-u,$$

转第 1 步.

算法 7.15 可以在连通图 G 的关联矩阵 A 上进行. 这是由于覆盖可用关联矩阵刻画：$K \subseteq V(G)$ 是覆盖的充要条件是，A 相应于 K 的顶点的行所构成的子矩阵，其每列都有元素 1.

算法 7.16　用关联矩阵实现算法 7.15 求最小覆盖的算法

设 G 是连通图，A 是关联矩阵，用 K 表示所求最小覆盖. 令 $K=\varnothing$.

第 1 步　取 $v_i \in V(G)$，使 A 的 v_i 行各元素的和最大，并令 $K=K\cup\{v_i\}$（当 v_i 的选取不唯一时，可能导致不同的结果，因此，必要时应比较所有的结果）.

第 2 步　在 A 划去 v_i 行及该行元素 1 所在的列.

第 3 步　若 A 的列划完，终止. 否则把剩下的矩阵仍记为 A，转第 1 步.

例 7.21　对图 7.15 的图 G 求最小覆盖. 图 G 是按下面的方法画出的. 任画 6 条边 (c_i,b_i)；6 个 b 点划分为 3 对，每对顶点各与 1 个 a 点相连；再把 6 个 b 点分成 2 个 3 点组，每个 3 点组各与 1 个 a 点相连；依次对 4 点组，5 点组重复同样的方法与 a 点相连.

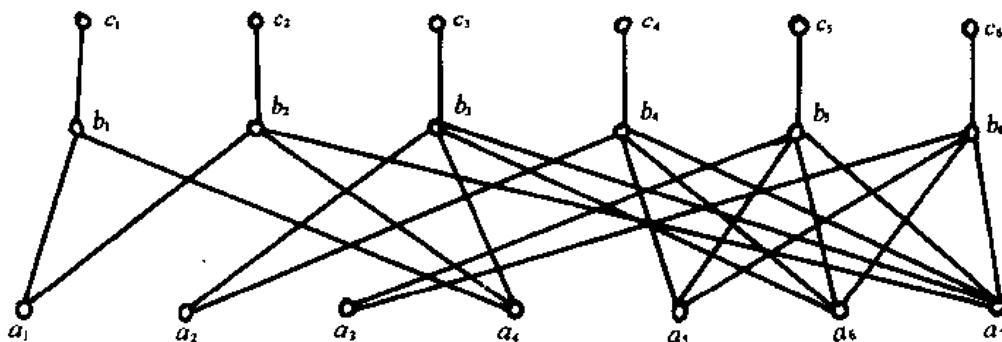


图 7.15

按每次取顶点度最大的顶点，可依次选取：

$$a_7,\ a_6,\ a_5,\ a_4,\ a_3,\ a_2,\ a_1,\ c_1,\ c_2,\ c_3,\ c_4,\ c_5,\ c_6,$$

故得 $K=\{a_1,a_2,a_3,a_4,a_5,a_6,a_7,c_1,c_2,c_3,c_4,c_5,c_6\}$. 另一方面我们看到，最小覆盖是

$$\{b_1,b_2,b_3,b_4,b_5,b_6\},$$

如果把 6 条边扩大为 n 条边 (c_i,b_i)，$i=1,\cdots,n$，则 a 点个数是

$$L(n) \triangleq \sum_{i=2}^{n-1}\left[\frac{n}{i}\right]. \tag{7.39}$$

这样的图的最小覆盖由 n 个 b 点构成，而由算法 7.15，可能产生 $L(n)+n$ 个顶点的覆盖，即增加了 $L(n)$ 个顶点，其增加比例是 $\frac{L(n)}{n}$，如表 7.6.

表 7.6

n	6	10	30	100	1000	…
$\frac{L(n)}{n}$ (%)	117	160	267	380	600	…

从表 7.6 知算法 7.15 所产生的覆盖，与最小覆盖所含顶点数相比，其顶点数增加的比例没有上界. 下面的算法所产生的覆盖，与最小覆盖所含顶点数相比，其顶点数增加的比例至多 100%.

算法 7.17　最小覆盖的启发式算法 2

设 G 是连通图，用 K 表示所求最小覆盖. 令 $K=\varnothing$.

第 1 步　若 $E(G)=\varnothing$，停. 否则取 $(u,v)\in E(G)$.

第 2 步　令 $K=K\cup\{u,v\}$，

$$G = G - \{u,v\},$$

转第1步.

在算法中所选定的边都不相邻，故每条边都必由覆盖中的不同顶点来覆盖. 因此，任何覆盖都含每条所选定的边的一个端点，从而任何覆盖至少含所选顶点的一半. 于是算法所产生的覆盖与最小覆盖所含顶点数相比，其顶点数增加的比例至多100%.

算法7.17也能在关联矩阵上实现(习题7.42).

§7.11 系统监控问题之二

例7.22 图7.16是一个指挥系统，顶点表示被指挥的单位，边表示可以直接下达命令的通讯线路. 欲在某些单位建立指挥站，以便可以通过指挥站直接给各单位下达命令，问如何建立指挥站使指挥站尽可能少?

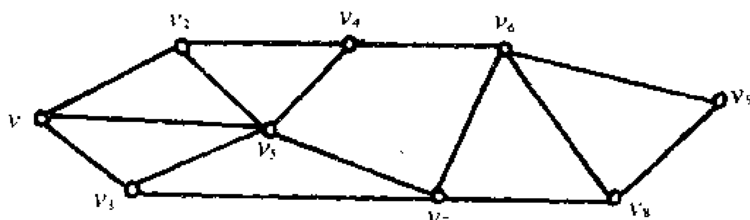


图7.16

假如在顶点v_1建立指挥站，则可通过这个指挥站直接给在v_1,v_2,v_3与v_5的单位下达命令. 因此，例7.22的问题就是在一个监控系统中"顶点监控顶点"的问题. 换言之，需在图7.16中找一个顶点子集$D\subseteq V(G)$. 对每个$v\in V(G)$，或者v在D中，或者v与D中某顶点相邻.

定义7.8 设G是连通图，$D\subseteq V(G)$，若对每个$v\in V(G)$，或$v\in D$，或$D\cap\Gamma(v)\neq\varnothing$，则称$D$为图$G$的一个**控制集**. 若$D$是控制集，而$D$的任何真子集都不是控制集，则称$D$是**极小控制集**. 顶点数最少的控制集称为**最小控制集**. 最小控制集所含顶点数目称为**控制数**，记为$\gamma_0(G)$或γ_0.

由定义7.8，例7.22的问题就是求最小控制集的问题.

定义7.9 设H是连通图G的邻接矩阵，I是单位矩阵，则$X\triangleq H+I$称为**控制矩阵**.

由定义7.8与定义7.9直接得到:

引理7.24 设X是连通图G的控制矩阵，则$D\subseteq V(G)$是控制集的充要条件是，X相应于D的顶点的行所构成的子矩阵，其每列都有元素1.

从引理7.24可得下列启发式算法.

算法7.18 最小控制集的启发式算法

设X是连通图G的控制矩阵，用D表示所求最小控制集. 令$D=\varnothing$.

第1步 取$v_i\in V(G)$，使X的v_i行各元素的和最大，并令$D=D\cup\{v_i\}$.

第2步 在X划去v_i行及该行元素1所在的列.

第 3 步　若 X 的列划完，终止. 否则把剩下的矩阵仍记为 X，转第 1 步.

例 7.23　求图 7.16 的最小控制集.

首先写出控制矩阵 X，

$$X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

取 $D = \{v_5\}$，划去第 5 行及第 1 至第 5 列，划去第 7 列，得

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccc} 6 & 8 & 9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array},$$

再取顶点 v_6，X 的列全划掉，终止，得

$$D = \{v_5, v_6\},$$

即在 v_5 与 v_6 建立两个指挥站就可以指挥系统中所有单位. 容易看到，一个指挥站是不行的，故所求为最小控制集. 控制数 $\gamma_0 = 2$.

求最小控制集可用逻辑算法.

引理 7.25　设 G 是连通图，则 $D \subseteq V(G)$ 是最小控制集的充要条件是，对每个顶点 $v \in V(G)$，恰有下列条件之一成立：

(1) $v \in D$，

(2) $D \cap \Gamma(v) \neq \varnothing$，

(习题 7.50).

引理 7.25 提出了求最小控制集的逻辑规则：

$$\text{对每个顶点 } v\text{，选择 } v \text{ 或选择 } v \text{ 的一个相邻顶点.} \tag{7.40}$$

由逻辑规则(7.40)可得逻辑算法.

算法 7.19　最小控制集的逻辑算法

设 G 是连通图.

第 1 步　由规则(7.40) 建立逻辑式.

第 2 步　由引理 7.8 的逻辑运算性质化简第 1 步的逻辑式.

第 3 步　由第 2 步直接写出全部极小控制集.

第 4 步　在第 3 步中顶点数最少的即为最小控制集.

习　题　七

7.1　证明推论 7.2.

7.2　若 G 是欧拉图,则 G 的块也是欧拉图.

7.3　连通有向图 $G=(V,E)$ 有有向欧拉闭迹的充要条件是

$$d^{+}(v)=d^{-}(v),\ v\in V.$$

7.4　用算法 7.1 求图 7.1(a) 的欧拉闭迹.

7.5　设 G 是欧拉图,$v_0\in V(G)$. 今从 v_0 出发沿 G 的边旅行,遵循如下规则:

(1) 沿没有经过的边旅行,

(2) 在从 G 删去已经旅行过的边的子图中,只要有不是桥的边则一定不过桥,

(3) 当旅行无法继续时终止.

证明当旅行终止时所得为欧拉闭迹.

7.6　按习题 7.5 的方法求图 7.1(a) 的欧拉闭迹.

7.7　图 7.17 为某城街道模拟图,试给边定向成为强连通图.

7.8　证明下列各条等价:

(1) G 有定向图是强连通的;

(2) G 连通且没有桥;

(3) G 是 2 边连通的;

(4) G 的任二顶点至少有两条边不交的路相连通.

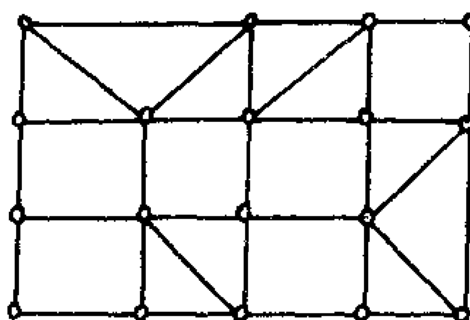


图 7.17

7.9　证明引理 7.6.

7.10　证明引理 7.7.

7.11　证明引理 7.8.

7.12　证明引理 7.9.

7.13　把式(7.18) 化简为式(7.19).

7.14　证明非平凡树的色数是 2.

7.15　设 G 非平凡图,则 $\chi(G)=2$ 的充要条件是 G 没有奇圈.

7.16　证明 $\chi(G)\leqslant\Delta(G)+1$.

7.17　设有 8 个犯人,为防止他们串供,必须把有牵连的犯人互相隔离,问至少需要几个关押室,已知有牵连的情况如表 7.7.

表 7.7

犯人	有牵连的犯人
A	B,C,E,G
B	A,C,H
C	A,B,D
D	C,E,H
E	A,D,F,H
F	E,G
G	A,F,H
H	B,D,E,G

7.18　证明引理 7.11.

7.19　证明引理 7.12.

7.20　证明引理 7.13.

7.21　对完全偶图 $K_{m,n}$，给出一个正常 Δ 边着色，从而证明 $\chi' = \Delta$.

7.22　有 4 位教师给 5 个培训班上课，授课要求如表 7.8，试安排一张课时数最少的课表.

7.23　设 G 是简单图，证明 $\chi' = \Delta$ 或 $\Delta + 1$.

7.24　设 G 是简单图，试给出一个正常 $(\Delta + 1)$ 边着色的好算法.

7.25　设 G 是 $\delta > 1$ 的简单图，试给出 $(\delta - 1)$ 边着色使

$$c(v) = \delta - 1,\ v \in V(G),$$

其中 $c(v)$ 是顶点 v 的色度.

表 7.8

课时＼班 教师	Y_1	Y_2	Y_3	Y_4	Y_5
X_1	2	1	2	3	0
X_2	0	3	1	1	0
X_3	0	0	2	2	1
X_4	0	0	2	0	4

7.26　证明引理 7.15.

7.27　对于习题 7.22 的排课表问题，

(1) 你安排了几间教室？

(2) 能否减少教室而课时不变？

(3) 如何安排只要 3 间教室的课表？

7.28　证明引理 7.17.

7.29　证明引理 7.18.

7.30　图 7.18 是中国邮递员问题的模拟图，方框 □ 表邮局，试给出邮递员从邮局出发，投递完邮件后又回到邮局的总路程最短的投递路线.

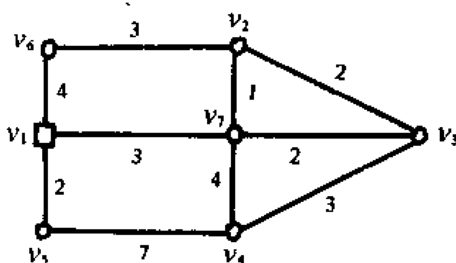


图 7.18

7.31　证明引理 7.20.

7.32　假定有 n 个工件在同一台机床加工，相继两个工件加工的中间机床有一个调整时间，设 t_{ij} 为工件 i 到工件 j 的调整时间，由此有一个 n 阶调整矩阵 $T = (t_{ij})_{n\times n}$. 如何给 n 个工件排出一个加工顺序，使总的调整时间最少？当 $n = 6$，设有 6 个工件，调整矩阵是

$$\begin{bmatrix} 0 & 5 & 3 & 4 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 2 \\ 2 & 5 & 0 & 1 & 2 & 3 \\ 1 & 4 & 4 & 0 & 1 & 2 \\ 1 & 3 & 4 & 5 & 0 & 5 \\ 4 & 4 & 2 & 3 & 1 & 0 \end{bmatrix},$$

试给出一个调整时间最少或近似最少的工件加工顺序.

7.33　给出竞赛图的邻接矩阵 H 如下，求有向哈密顿路.

$$H=\begin{bmatrix}0&0&1&1&0&0&1\\1&0&1&0&1&0&0\\0&0&0&1&1&1&0\\0&1&0&0&1&1&1\\1&0&0&0&0&0&1\\1&1&0&0&1&0&1\\0&1&1&0&0&0&0\end{bmatrix}$$

7.34　证明引理 7.21.

7.35　证明定理 7.22.

7.36　证明定理 7.23.

7.37　(1) 画出 4 个顶点的竞赛图，

(2) 对所有 4 个顶点的竞赛图分别排名.

7.38　给出竞赛图 G 的邻接矩阵 H，试排名.

$$H=\begin{bmatrix}0&1&1&1&1&1&0&0&1&1\\0&0&1&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&1&1&0&1&1&0&0&1&0\\0&1&1&0&0&0&0&0&0&0\\0&0&1&0&1&0&0&0&1&0\\1&1&1&1&1&1&0&0&1&0\\1&1&1&1&1&1&1&0&1&1\\0&1&1&0&1&0&0&0&0&0\\0&1&1&1&1&1&1&0&1&0\end{bmatrix}$$

7.39　给定旅行推销员问题的权矩阵

$$W=\begin{array}{c}\\A\\B\\C\\D\\E\\F\end{array}\begin{array}{c}\begin{array}{cccccc}A&B&C&D&E&F\end{array}\\\begin{pmatrix}0&56&35&2&51&60\\56&0&21&57&78&70\\35&21&0&36&68&68\\2&57&36&0&51&61\\51&78&68&51&0&13\\60&70&68&61&13&0\end{pmatrix}\end{array},$$

试由算法 7.13 求最小权哈密顿圈并给出其下界，设初始哈密顿圈是 $C = ABCDEFA$.

7.40　用算法 7.14 重解习题 7.39 的旅行推销员问题.

7.41　若在街区的交叉路口设置邮筒，则与该路口紧相通的街道都能使用所设置的邮筒，问至少在该街区设置多少邮筒才能保证每条街道的居民不用跨过交叉路口而使用邮筒?试对习题 7.30 的街道图给出具体结果.

7.42　如何在关联矩阵上实现算法 7.17?

7.43　用算法 7.18 能求出图 7.15 的最小控制集吗?

7.44　中国象棋的马，能否在中国象棋盘上从某一方格出发，正好跳达每个格点一次？为什么？

7.45　中国象棋的马，能否在国际象棋盘上从某一格点出发，正好跳达每个方格一次最后回到出发的方格？为什么？

7.46　已知 A,B,C,D,E,F 与 G 共 7 个人，A 会讲英语；B 会讲英语与汉语；C 会讲英语，意大利语与俄语；D 会讲汉语与日语；E 会讲意大利语与德语；F 会讲俄语，日语与法语；G 会讲德语与法语. 能否安排这 7 个人坐在一张圆桌旁，使每个人都能与自己身边的两个人交谈？

7.47　某专业有 9 个研究生，其选修课程如表 7.9.

表 7.9

研究生	1	2	3	4	5	6	7	8	9
选修课程	A	A	D	A	C	B	A	A	C
	B	B	E	F	F	D	C	E	E
	C	D				F	F		

问在期末考试时，至少要分几场考试，并作出考试安排.

7.48　证明极大独立集必是极小控制集，但其逆不成立.

7.49　证明控制数不超过独立数，即

$$\gamma_0(G) \leqslant \beta_0(G).$$

7.50　证明引理 7.25.

7.51　求图 7.16 的最小控制集，用算法 7.19.

7.52　图 7.19 是某城市的部分街道图，一辆清扫车停在车库 ⊗ 的位置. 图中箭头所指为单行街道，其余街道皆为双向行驶. 假定清扫车从车库出发，清扫全部街道后回到车库，试给出一条经济的清扫路线.

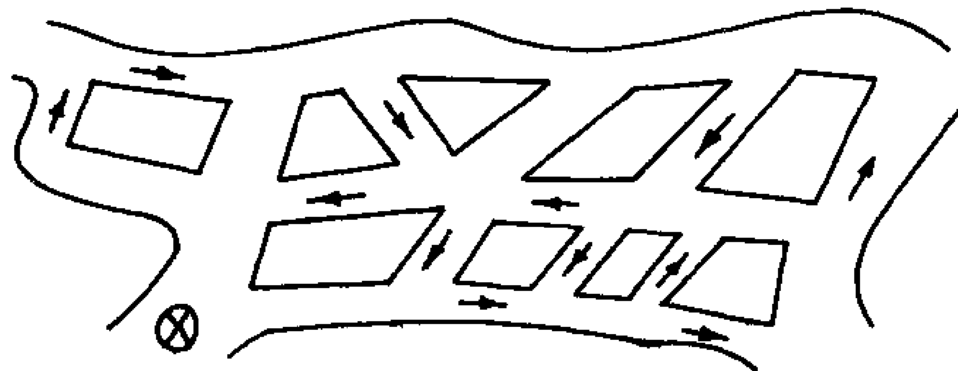


图 7.19

7.53　设图 $G=(V,E)$ 是中国邮递员问题的非负权连通图，G 不是欧拉图，E' 是复制边集使

$$G'=(V,E\cup E')$$

是欧拉图，证明 E' 是最小权复制边集的充要条件是

(1) E' 中没有平行边，

(2) G 中每一个圈上所有复制边的权和不超过该圈权的一半.

7.54　解中国邮递员问题首先是找出所有奇顶点，然后附加复制边使其都成为偶顶点，再由习题 7.53 在图的每个圈上进行调整.因此这样的方法称为奇偶点图上作业法.试给出计算步骤.

7.55　利用习题 7.54 的计算步骤，重解习题 7.30 的中国邮递员问题.

第八章　网络及其应用

§8.1　网络和网络流

网络问题在前几章中已经遇到，例如最短路问题，最小生成树问题，匹配问题，中国邮递员问题，旅行推销员问题等. 本章将对网络问题作进一步讨论.

定义 8.1　称 $N=(V,E,c,X,Y)$ 为一个网络，如果

(1) $G=(V,E)$ 是一个有向图，

(2) c 是 E 上正整数函数，称为**容量函数**，对每条边 e，$c(e)$ 称为边 e 的容量，

(3) X 与 Y 是 V 的两个非空不相交子集，分别称为 G 的**发点集**与**收点集**，$I=V\backslash(X\cup Y)$ 称为 G 的**中间点集**. X 的顶点称为**发点**或**源**，Y 的顶点称为**收点**或**汇**，I 的顶点称为**中间点**.

在一个网络中，通过每条边的流量可看成边集 E 上的一个函数，它受两种限制：

(1) 边的流量不能超过边的最大输送能力，即流量不能超过容量，

(2) 在每个中间点，流进与流出该顶点的总流量相等，即保持中间点的流量平衡.

由此我们有下面的定义.

定义 8.2　设 N 是一个网络，f 是 E 上整数值函数，如果

(1) $0\leqslant f(e)\leqslant c(e),\ e\in E,$　　(8.1)

(2) $\sum\limits_{e\in N^+(v)} f(e)=\sum\limits_{e\in N^-(v)} f(e),\ v\in I,$　　(8.2)

则称 f 是网络 N 的一个**流**. $f(e)$ 是边 e 的**流量**. 条件(8.1)叫**容量限制**. 条件(8.2)叫**守恒条件**.

例 8.1　如图 8.1 的网络有 2 个发点 x_1 与 x_2；3 个收点 y_1,y_2 与 y_3；4 个中间点 v_1,v_2,v_3 与 v_4. 在箭杆旁的 2 个数，左边是流量，右边是容量.

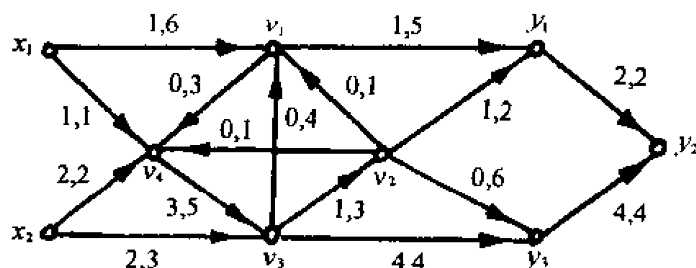


图 8.1　网络与网络流

设 $A \subseteq V$，引入记号

$$f^+(A) \triangleq \sum_{e \in N^+(A)} f(e), \tag{8.3}$$

$$f^-(A) \triangleq \sum_{e \in N^-(A)} f(e), \tag{8.4}$$

则守恒条件(8.2)可写为

$$f^+(v) = f^-(v), \ v \in I. \tag{8.5}$$

定义 8.3 设 f 是网络 N 的一个流，$A \subseteq V$，则称 $f^+(A) - f^-(A)$ 为流出 A 的合成流量，$f^-(A) - f^+(A)$ 为流入 A 的合成流量.

引理 8.1 设 f 是网络 N 的一个流，则

$$f^+(X) - f^-(X) = f^-(Y) - f^+(Y), \tag{8.6}$$

即流出发点集 X 的合成流量等于流入收点集 Y 的合成流量(习题 8.2).

由引理 8.1 得下面的定义.

定义 8.4 设 f 是网络 N 的一个流，则 f 的流值 val f 定义为

$$\text{val}\, f = f^+(X) - f^-(X), \tag{8.7}$$

即流值是发点集的流出量，也是收点集的流入量.

例 8.2 图 8.1 中网络 N 的流 f，其流值为

$$\begin{aligned}\text{val}\, f &= f(x_1,v_1) + f(x_1,v_4) + f(x_2,v_3) + f(x_2,v_4) \\ &= f(v_1,y_1) + f(v_2,y_1) + f(v_2,y_3) + f(v_3,y_3) = 6.\end{aligned}$$

任何一个网络 $N = (V,E,c,X,Y)$，都等价于一个发点和一个收点的网络

$$N' = (V',E',c',s,t),$$

(1) $V' = V \cup \{s,t\}$，s 与 t 分别是 N' 的发点与收点，

(2) $E' = E \cup \{(s,x) \mid x \in X\} \cup \{(y,t) \mid y \in Y\}$，

(3) $c'(e) = c(e),\ e \in E;\ c'(s,x) = \infty,\ x \in X;\ c'(y,t) = \infty,\ y \in Y.$

例 8.3 图 8.1 的网络等价于图 8.2 的网络. 图中箭杆旁的数是(f,c').

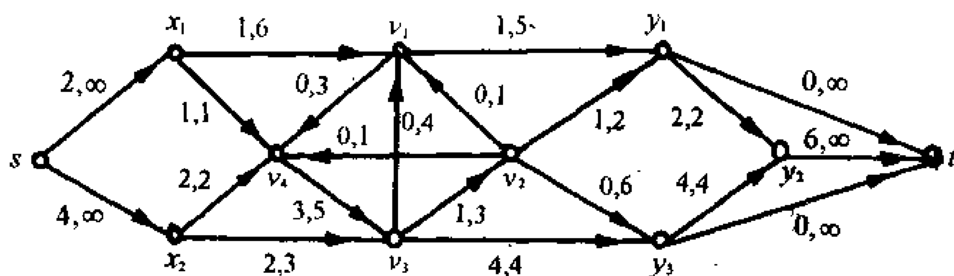


图 8.2 一个发点与一个收点的网络

对于网络 $N = (V,E,c,s,t)$，显然有

$$\text{val}\, f = f^+(s) = f^-(t).$$

在图 8.2 中，$\text{val}\, f = f(s,x_1) + f(s,x_2)$

$$= f(y_1,t) + f(y_2,t) + f(y_3,t) = 6.$$

§8.2　最大流和最小割

定义 8.5　设 $N=(V,E,c,s,t)$ 是一个网络，f 是一个流，若不存在流 f' 使

$$\text{val } f' > \text{val } f,$$

则称 f 为 N 的**最大流**.

若 $A\subseteq V, s\in A, t\in \overline{A} \triangleq V\backslash A$，则

$$(A,\overline{A}) \triangleq N^+(A)$$

称为网络 N 的一个**割**，而

$$\text{cap }(A,\overline{A}) \triangleq \sum\{c(e)\mid e\in (A,\overline{A})\}$$

称为割 $(A,\overline{A})$ 的**容量**. K 是一个割，若不存在割 K' 使

$$\text{cap } K' < \text{cap } K,$$

则称 K 为 N 的**最小割**.

定理 8.2　设 f 是网络 N 的流，$(A,\overline{A})$ 是一个割，则

(1) $\text{val } f = f^+(A) - f^-(A)$,　　(8.8)

(2) $\text{val } f \leqslant \text{cap }(A,\overline{A})$.　　(8.9)

证　由定义，

$$f^+(v) - f^-(v) = \begin{cases} \text{val } f, & v = s, \\ 0, & v\in A\backslash\{s\}, \end{cases}$$

在 A 上求和，

$$\text{val } f = \sum_{v\in A}(f^+(v) - f^-(v)),$$

再由习题 8.1 即得式(8.8).

由式(8.8)，

$$\begin{aligned} \text{val } f &= \sum_{e\in N^+(A)} f(e) - \sum_{e\in N^-(A)} f(e) \\ &\leqslant \sum_{e\in N^+(A)} f(e) \\ &\leqslant \sum_{e\in N^+(A)} c(e) \\ &= \text{cap }(A,\overline{A}), \end{aligned}$$

故式(8.9)得证.　**证毕.**

定理 8.3　设 f 是流，K 是割，若

$$\text{val } f = \text{cap } K, \tag{8.10}$$

则 f 是最大流，K 是最小割.

证　设 f^* 是最大流，K^* 是最小割，则

$$\text{val } f \leqslant \text{val } f^* \leqslant \text{cap } K^* \leqslant \text{cap } K,$$

故得证.

例 8.4　对图 8.3 的网络 N 共有 8 个割，如表 8.1.

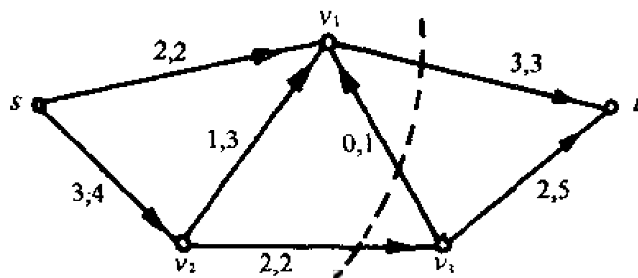


图 8.3 最大流和最小割

表 8.1 网络的割

A	$(A,\bar{A})$	cap $(A,\bar{A})$
$\{s\}$	$(s,v_1),(s,v_2)$	6
$\{s,v_1\}$	$(s,v_2),(v_1,t)$	7
$\{s,v_2\}$	$(s,v_1),(v_2,v_1),(v_2,v_3)$	7
$\{s,v_3\}$	$(s,v_1),(s,v_2),(v_3,v_1),(v_3,t)$	12
$\{s,v_1,v_2\}$	$(v_2,v_3),(v_1,t)$	5*
$\{s,v_1,v_3\}$	$(s,v_2),(v_1,t),(v_3,t)$	12
$\{s,v_2,v_3\}$	$(s,v_1),(v_2,v_1),(v_3,v_1),(v_3,t)$	11
$\{s,v_1,v_2,v_3\}$	$(v_1,t),(v_3,t)$	8

在表8.1中打“*”者为最小割容量. 图8.3中的流的流值为5,等于最小割容量,故它是最大流.

下面将证明定理 8.3 的逆定理亦成立,即网络的最大流值与最小割容量必相等.

引理 8.4 设 $N=(V,E,c,s,t)$ 是一个网络,$E=\{1,2,\cdots,m\}$,今作 $G^0=(V,E^0)$,$E^0=\{0,1,\cdots,m\}$,边 0 表示边(t,s) 并称为返回边,其容量为 ∞,则求 N 的最大流等价于求 G^0 的满足容量限制的环流并使返回边的流量尽可能大(习题 8.11).

定理 8.5(最大流 — 最小割定理) 网络 N 的最大流值等于最小割容量.

证 设 $\varphi^0=(\varphi_0,\varphi_1,\cdots,\varphi_m)^T$ 是 G^0 的满足容量限制的环流且 φ_0 尽可能大. 给 G^0 的边如下着色:

(1) 返回边着黑色,

(2) $\varphi_i=0$ 的边 i 着黑色,

(3) $0<\varphi_i<c_i$ 的边 i 着红色,

(4) $\varphi_i=c_i$ 的边 i 着绿色.

若出现定理 1.14 的情形 1,即有圈 C 含返回边,不含未着色边,且黑边与返回边同向,绿边与返回边反向. 设圈 C 相应的圈向量是 μ,不妨设返回边 0 的分量为 $+1$,即

$$\begin{matrix} & \text{返回边} & \text{黑边} & \text{绿边} & \text{红边} \\ \mu=(& 1 \ \cdots & 1 \ \cdots & -1\cdots & \pm 1\cdots)^T, \end{matrix}$$

于是 $\varphi^0+\mu$ 是 G^0 的满足容量限制的环流且返回边的流量更大,矛盾. 因此必然出现定理 1.14 的情形 2,即有含返回边的余圈$[A,\bar{A}]$,余圈不含红边,黑边与返回边同向,绿边与返

回边反向．不妨设 $s \in A, t \in \overline{A}$，则有

$$N^+(A) = \{e \in [A, \overline{A}] \mid e \text{ 绿边}\},$$
$$N^-(A) = \{e \in [A, \overline{A}] \mid e \text{ 黑边}\},$$

由引理 8.4，$\varphi = (\varphi_1, \cdots, \varphi_m)^T$ 是 N 的流且

$$\begin{aligned} \operatorname{val} \varphi = \varphi_0 &= \sum_{i \in N^+(A)} \varphi_i - \sum_{i \in N^-(A)} \varphi_i \\ &= \sum_{i \in N^+(A)} c_i = \operatorname{cap} N^+(A), \end{aligned}$$

再由定理 8.3，φ 是最大流，$N^+(A)$ 是最小割．　**证毕．**

§8.3　应用——最大流问题

例 8.5　由煤矿 s 通过图 8.4(a) 的网络运送煤到城市 t，问最大运送量是多少(单位百万吨)？

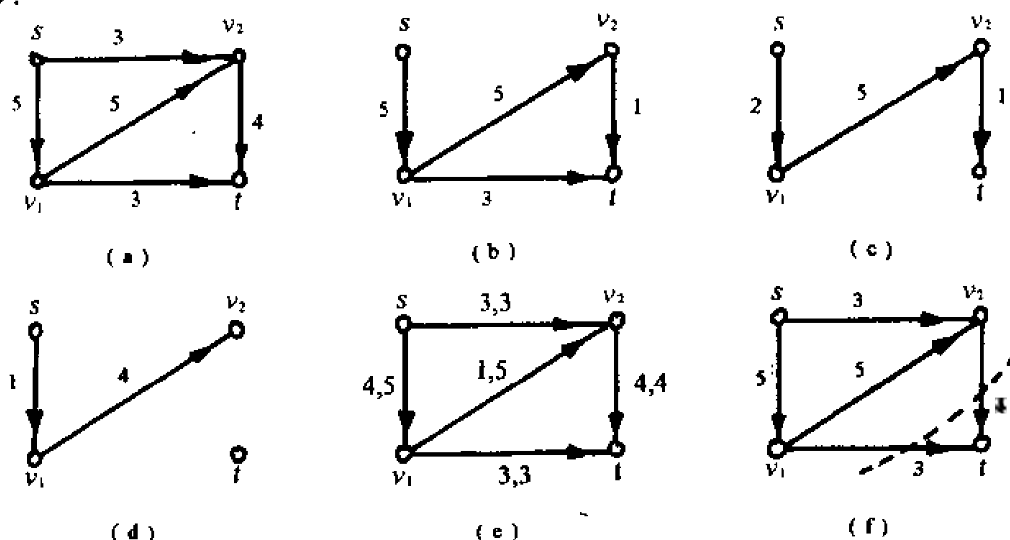


图 8.4　(a) 网络　(b) ~ (e) 调整过程　(f) 最小割

从零流开始，在图 8.4(a) 中任选 s-t 路 P，

$$P = sv_2t,$$

沿 P 从 s 到 t 可增加运送流量 3，于是得(b)．

在(b)，沿 s-t 路 $P = sv_1t$ 可增加运送流量 3，于是得(c)．

在(c)，沿 s-t 路 $P = sv_1v_2t$ 可增加运送流量 1，于是得(d)．

在(d)，没有 s-t 路，不可能再增加运送流量，算法过程终止．

综合(a) ~ (d) 的过程，得流 f 如(e) 所示，其流值为 7．

从(f)，有一个割的容量是 7，故由定理 8.3，所得到的流是最大流，即从 s 到 t 的最大运送量是 7 百万吨．

从例 8.5 看到，求最大流的一个启发式算法是从网络的任一流开始，再沿 s-t 路 P 尽可能增加流量，当不能再增加流量时终止．因此称这样的算法为单向调整法．我们还看到，沿 s-t 路 P 增加运送的最大流量就是 P 的边的最小容量，称为 P 的容量．

定义 8.6 设 P 是网络 N 的 s-t 路，则

$$c(P) = \min\{c(e) \mid e \in E(P)\} \tag{8.11}$$

称为 P 的容量.

算法 8.1 最大流的单向调整法

设网络 $N = (V, E, c, s, t)$，流 $f = 0$.

第 1 步　若没有 s-t 路，终止. f 为所求.

第 2 步　任取 s-t 路 P，则沿 P 可增加运送流量 $c(P)$ 而得新的流，即令

$$f(e) = \begin{cases} f(e) + c(P), & e \in E(P), \\ f(e) & \text{，否则}. \end{cases}$$

$$c(e) = \begin{cases} c(e) - c(P), & e \in E(P), \\ c(e) & \text{，否则}. \end{cases}$$

第 3 步　令 $\overline{E} = \{e \mid c(e) > 0\}$.

第 4 步　令 $E = \overline{E}$，返回第 1 步.

算法 8.1 称为单向调整，是因为只考虑沿 s-t 路尽可能多地增加运送流量. 这对局部来说是最好的，但对整体来说未必是最好的，如下面的例.

例 8.6 按单向调整法重解图 8.4(a) 的网络. 任选 s-t 路

$$P = sv_1v_2t,$$

$c(P) = 4$，经单向调整即沿 P 尽可能增加运送流量，如图 8.5(a).

在(a)，沿 s-t 路 $P = sv_1t$ 单向调整得(b).

在(b)，不存在 s-t 路，终止，得流 f 如(c)，其流值为 5，显然这不是最大流.

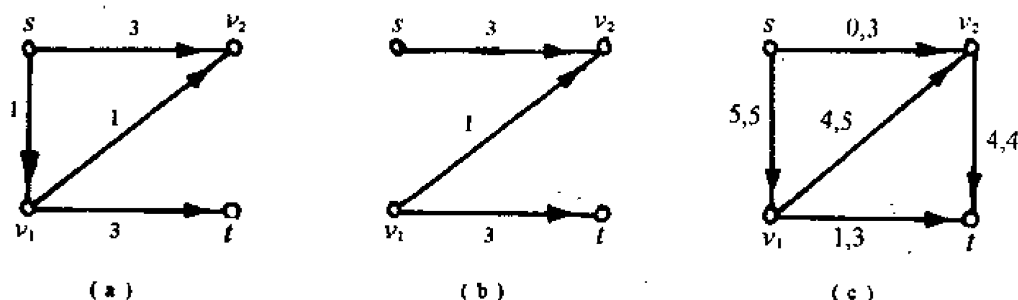


图 8.5　单向调整过程

单向调整未必求出最大流，如例 8.6. 其原因在于 s-t 路不唯一，而一旦选定 s-t 路 P，则尽可能沿 P 多运送流量. 把单向调整改为双向调整即可克服这一缺点. 所谓**双向调整法**即沿 s-t 路 P 增加运送流量 $c(P)$ 得到新的流 f' 之后，对每条边 $e \in E(P)$，意味着沿 e 存在双向选择：沿 e 的方向可增加运送流量为 $c(e) - f'(e)$，沿 e 的反方向可运送流量为 $f'(e)$，即沿 e 的方向可减少运送流量为 $f'(e)$.

例 8.7 按双向调整法重解图 8.4(a) 的网络.

同例 8.6 一样，选 s-t 路 $P = sv_1v_2t$，由双向选择原则得图 8.6(a). 图中虚边表示反向流动的边.

在(a)，同例 8.6 一样，选 s-t 路 $P = sv_1t$，由双向选择原则得(b).

在(b)有 s-t 路 $P = sv_2v_1t$，由于边(v_2,v_1)是虚边，故沿(v_2,v_1)运送 $c(P) = 2$，即沿(v_1,v_2)减少运送 $c(P) = 2$，由此得(c).

在(c)不存在 s-t 路，终止，得流 f 如(d)，$\mathrm{val}\ f = 7$. 由例 8.5 知 f 是最大流.

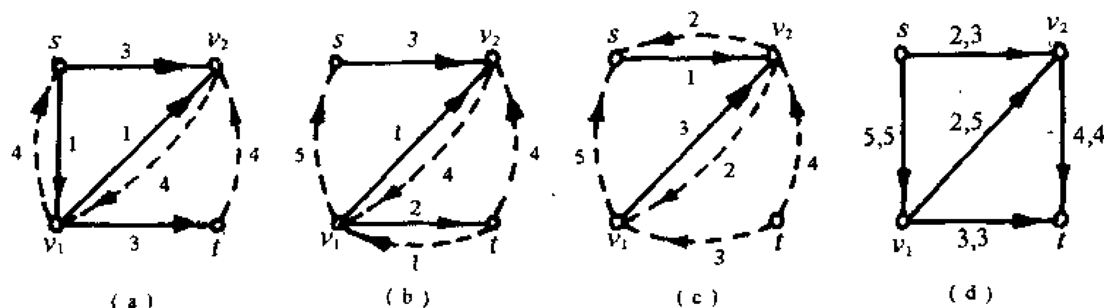


图 8.6　双向调整过程

注意到对图 8.4(a)的网络的双向调整求得最大流，下面证明双向调整必能求得最大流.

定义 8.7　设 f 是网络 $N = (V,E,c,s,t)$ 的一个流，网络 $N(f) = (V,E^0,c^0,s,t)$ 称为 N **关于流 f 的伴随网络**，如果

(1) $E^0 = E^0_+ \cup E^0_-$,

(2) 对每条边$(u,v) \in E$,

若 $f(u,v) = 0$，则

$$(u,v) \in E^0_+,\ c^0(u,v) = c(u,v);$$

若 $f(u,v) = c(u,v)$，则

$$(v,u) \in E^0_-,\ c^0(v,u) = c(u,v);$$

若 $0 < f(u,v) < c(u,v)$，则

$$(u,v) \in E^0_+,\ c^0(u,v) = c(u,v) - f(u,v),$$

$$(v,u) \in E^0_-,\ c^0(v,u) = f(u,v).$$

并称 E^0_+ 的边为**顺向边**，E^0_- 的边为**反向边**.

由定义，若$(u,v) \in E^0_+$，则 $c^0(u,v)$ 表示沿顺向边(u,v)可增加运送的最大流量. 若$(v,u) \in E^0_-$，则 $c^0(v,u)$ 表示沿反向边(v,u)可增加运送的最大流量，即沿原网络 N 的边(u,v)可减少运送的最大流量.

引理 8.6　设 $N(f)$ 是网络 N 关于流 f 的伴随网络，P 是 $N(f)$ 中的 s-t 路，δ 是不超过 $c^0(P)$ 的正整数，令

$$\hat{f}_\delta(u,v) = \begin{cases} f(u,v) + \delta, & (u,v) \in E(P) \cap E^0_+, \\ f(u,v) - \delta, & (v,u) \in E(P) \cap E^0_-, \\ f(u,v) & ，其它, \end{cases} \tag{8.12}$$

则 $\hat{f}_\delta$ 是 N 的一个流(习题 8.14).

由引理 8.6 得下面的定义.

定义 8.8　由式(8.12)定义的 $\hat{f}_\delta$ 称为网络 N 的**流 f 关于 P 与 δ 的修改流**. 若取 $\delta = c^0(P)$，则记 $\hat{f}_\delta$ 为 $\hat{f}$，并称为**流 f 关于 P 的修改流**.

引理 8.7　设 $\hat{f}_\delta$ 是网络 N 的流 f 关于 P 与 δ 的修改流，则

$$\text{val}\ \hat{f}_\delta = \text{val}\ f + \delta, \tag{8.13}$$

(习题 8.15).

引理 8.8　设 f 与 $(A,\bar{A})$ 分别是网络 N 的流与割，则 $\text{val}\ f = \text{cap}\ (A,\bar{A})$ 的充要条件是在 $N(f)$ 中

$$(A,\bar{A}) = \varnothing.$$

(习题 8.16).

定理 8.9　f 是 N 的最大流的充要条件是，在 $N(f)$ 中不存在 s-t 路.

证　**必要性**　若 $N(f)$ 中有 s-t 路 P，则由式(8.13)，可得修改流 $\hat{f}$，$\text{val}\ \hat{f} > \text{val}\ f$，矛盾.

充分性　设 $N(f)$ 中不存在 s-t 路，令

$$A = \{v \mid \text{在 } N(f) \text{ 中有 } s\text{-}v \text{ 路}\},$$

则 $s \in A, t \in \bar{A}$ 且在 $N(f)$ 中 A 只有入边，即在 $N(f)$ 中 $(A,\bar{A}) = \varnothing$. 由引理 8.8，$\text{val}\ f = \text{cap}\ (A,\bar{A})$，再由定理 8.3，$f$ 是最大流.　**证毕.**

由定理 8.9 可得求最大流的双向调整方法. 双向调整是由伴随网络实现的. 在伴随网络 $N(f)$ 中从 s 生长一棵有序树 T，若 t 能生长到 T，则逆向追踪得 $N(f)$ 中的 s-t 路 P，再由式(8.12) 作修改流 $\hat{f}$，其流值增加 $c^0(P)$；若 t 不能生长到 T，则已得最大流.

算法 8.2　最大流的双向调整法—— Ford-Fulkerson 算法

设 $N = (V,E,c,s,t)$，f 是初始流.

T 表 $N(f)$ 中以 s 为根的有序树；R 表在有序树 T 中已生长完成的顶点，T 中每个顶点 v 如下获得 2 个标号 $l_1(v)$ 与 $l_2(v)$. 设 P 是 T 上唯一的 s-v 路，则 $l_1(v)$ 表示 P 的容量，$l_2(v)$ 表示在 P 上 v 的父顶点，即有

$$l_1(v) = c^0(P), \tag{8.14}$$

$$l_2(v) = \begin{cases} +u, & \text{当}(u,v) \in E(P) \cap E^0_+, \\ -u, & \text{当}(u,v) \in E(P) \cap E^0_-. \end{cases} \tag{8.15}$$

第 1 步　令 $T = \{s\}, l_1(s) = \infty,\ l_2(s) = s, R = \varnothing$.

第 2 步　若 $T \backslash R = \varnothing$，则 f 是最大流，$(T,\bar{T})$ 是最小割.

第 3 步　取 $T \backslash R$ 中第 1 个顶点 u 作为生长顶点.

第 4 步　在 $N(f)$ 中割 $(T,\bar{T})$ 不存在以 u 为起点的有向边，令 $R = R \cup \{u\}$，转第 2 步.

第 5 步　取 $(u,v) \in (T,\bar{T})$，令

$$T = T \cup \{v\},$$

$$l_1(v) = \min\ \{l_1(u), c^0(u,v)\},$$

$$l_2(v) = \begin{cases} +u, & (u,v) \in E^0_+, \\ -u, & (u,v) \in E^0_-. \end{cases}$$

第 6 步　若 $v \neq t$，转第 4 步.

第 7 步　逆向追踪求出 s-t 路 P.

第 8 步　对 $(u,v) \in E(P)$，令

$$
f(u,v)=\begin{cases} f(u,v)+l_1(t), & l_2(v)=+u, \\ f(u,v)-l_1(t), & l_2(v)=-u, \end{cases}
$$

转第 1 步.

关于算法的收敛性由下面的引理给出.

引理 8.10 算法 8.2 在有限步后求得最大流(习题 8.17).

例 8.8 用算法 8.2 求图 8.4(a) 的最大流,设初始流为零流. 其过程如图 8.7.

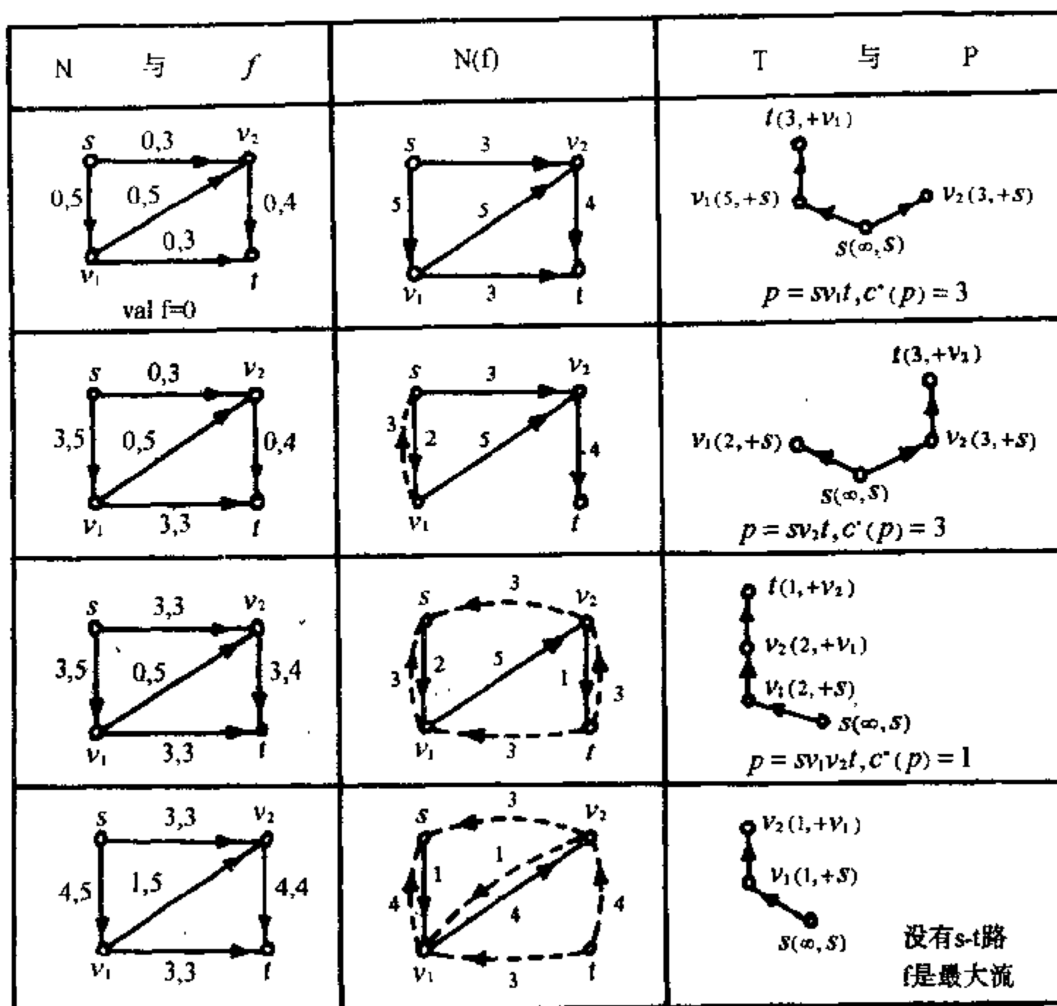


图 8.7 双向调整算法

网络 N 中箭杆旁的数字是(f,c),伴随网络 $N(f)$ 中箭杆旁的数字是 c^0

§8.4 应用——最小代价流问题

例 8.9 图 8.8(a) 的网络是附加代价的网络. 每条边有 2 个数对应,左边一个数表示容量,右边一个数表示代价. 边的代价表示沿该边的单位运价. 在图 8.8(b),(c) 与(d) 中分别给出流值为 4 的 3 个流,箭杆旁的 3 个数从左到右分别是边的流量,容量与代价. 对(b),(c) 与(d) 三个具有代价的网络,其总的代价分别是 28,24 与 15. 换言之,流值相同而代价却不同,问题是如何求代价最小的流?

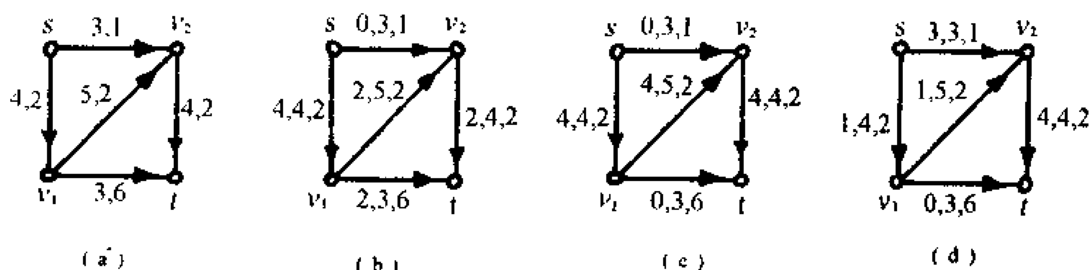


图 8.8　具代价的网络与网络流

定义 8.9　设 $N=(V,E,c,s,t)$ 是一个网络，若每条边 e 对应一个实数 $w(e)$，称为边 e 的代价，这样的网络称为**具代价的网络**，记为

$$N=(V,E,c,w,s,t).$$

若 f 是一个流，则

$$w(f)=\sum_{e\in E}f(e)w(e) \tag{8.16}$$

称为流 f 的代价. 若 f^* 是一个流，$\mathrm{val}\, f^*=\lambda$ 且

$$w(f^*)=\min\{w(f)\,|\,f\text{ 是流且 }\mathrm{val}\, f=\lambda\}, \tag{8.17}$$

则 f^* 称为**流值 λ 的最小代价流**. 若 λ 是最大流值，则称 f^* 为**最小代价最大流**.

用穷举法可以验证，图 8.8(d) 是流值 4 的最小代价流(习题 8.21).

定义 8.10　设 $N=(V,E,c,w,s,t)$，称

$$N(f)=(V,E^0,c^0,w^0,s,t)$$

为 N **关于流 f 的具代价的伴随网络**，如果

(1) (V,E^0,c^0,s,t) 是 (V,E,c,s,t) 的伴随网络.

(2) 对 E^0 的边的代价定义为

$$w^0(e)=\begin{cases}w(e), & e\in E_+^0,\\ -w(e), & e\in E_-^0.\end{cases} \tag{8.18}$$

定义 8.11　设 $N(f)$ 是具代价的伴随网络，Q 是 $N(f)$ 中的有向圈，则

$$c^0(Q)=\min\{c^0(e)\,|\,e\in E(Q)\} \tag{8.19}$$

称为 Q 的**容量**.

引理 8.11　设 $N(f)$ 是具代价的伴随网络，Q 是 $N(f)$ 中的有向圈，令

$$\tilde{f}_\delta(u,v)=\begin{cases}f(u,v)+\delta, & (u,v)\in E(Q)\cap E_+^0,\\ f(u,v)-\delta, & (v,u)\in E(Q)\cap E_-^0,\\ f(u,v) & ,\text{其它},\end{cases} \tag{8.20}$$

则 $\tilde{f}_\delta$ 是 N 的一个流(习题 8.22).

由引理 8.11 得下面的定义.

定义 8.12　由式(8.20)定义的 $\tilde{f}_\delta$ 称为**流 f 关于 Q 与 δ 的修改流**. 若取 $\delta=c^0(Q)$，则记 $\tilde{f}_\delta$ 为 $\tilde{f}$，并称为**流 f 关于 Q 的修改流**.

引理 8.12　设 $\tilde{f}$ 是流 f 关于 Q 与 δ 的修改流，则

$$\mathrm{val}\,\tilde{f}_\delta=\mathrm{val}\, f, \tag{8.21}$$

$$w(\tilde{f}_\delta)=w(f)+\delta w^0(Q), \tag{8.22}$$

(习题 8.23).

定理 8.13 设 f 是具代价的网络 N 的流，其流值为 λ，$N(f)$ 是具代价的伴随网络，则 f 是流值 λ 的最小代价流的充要条件是，$N(f)$ 中不存在有向圈 Q 使 $w^0(Q)<0$，即 $N(f)$ 中没有负回路(习题 8.24).

以上提供了求流值 λ 的最小代价流的方法. 先求出流值 λ 的流 f. 若 $N(f)$ 中没有负回路，则 f 是流值 λ 的最小代价流. 若 $N(f)$ 中有负回路 Q，作 f 关于 Q 的修改流 $\tilde{f}$，$\tilde{f}$ 保持流值不变而代价减少 $|c^0(Q)w^0(Q)|$. 用 $\tilde{f}$ 代替 f 重复迭代，直至求出最小代价流. 换言之，算法的基本思想是保持流值不变而逐步减少代价.

算法 8.3 最小代价流的负回路算法

设 $N=(V,E,c,w,s,t)$ 是具代价的网络.

第 1 步 求流值 λ 的流 f.

第 2 步 作具代价的伴随网络 $N(f)$.

第 3 步 若 $N(f)$ 没有负回路，f 是流值 λ 的最小代价流，停.

第 4 步 设 Q 是负回路，作 f 关于 Q 的修改流 $\tilde{f}$. 令 $f=\tilde{f}$，转第 2 步.

例 8.10 求图 8.8(a) 的流值 4 的最小代价流，设图 8.8(b) 的流为初始流. 其解的过程如图 8.9.

N 与 f	N(f)	Q
$W(f)=28$		$Q=v_1v_2tv_1$ $c^0(Q)=2$ $w^0(Q)=-2$ $c^0(Q)w^0(Q)=-4$
$W(f)=28-4=24$		$Q=sv_2v_1s$ $c^0(Q)=3$ $w^0(Q)=-3$ $c^0(Q)w^0(Q)=-9$
$W(f)=24-9=15$		没有负回路 f为流值4的 最小代价流

图 8.9 负回路算法

网络 N 中箭杆旁的数字是 (f,c,w)，具代价的伴随网络 $N(f)$ 中箭杆旁的数字是 (c^0,w^0)

定理 8.14 设 f 是 N 中流值 λ 的最小代价流，P 为 $N(f)$ 中最短 s-t 路(关于权 w^0)，δ 为不超过 $c^0(P)$ 的正整数，则 f 关于 P 与 δ 的修改流 $\hat{f}_\delta$ 是 N 中流值为$(\lambda+\delta)$ 的最小代价流.

证 由引理8.7和定理8.13，只需证在$N(\hat{f}_\delta)$ 中没有负回路. 设$N(\hat{f}_\delta)$ 中有负回路Q. 因为 $N(f)$ 不含负回路，而 $N(\hat{f}_\delta)$ 是从 $N(f)$ 中改变某些边而得到的，所以 Q 必含边 $e=(u,v)$，其代价为 $-w(e)$，而边(v,u) 是 P 的边，如图 8.10. 若用 $Q-e$ 代替 P 的边(v,u)，则从 s 到 t 的长是$(w(P)+w(Q))$，且满足

$$w(P)+w(Q)<w(P),$$

此与 P 为最短路矛盾. **证毕.**

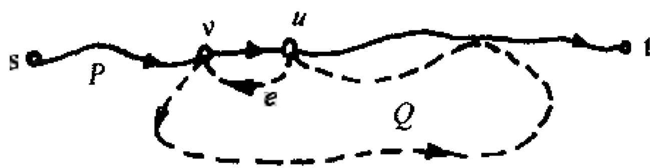


图 8.10

由此，求流值 λ 的最小代价流的另一方法是保持最小代价而逐步增加流值. 任取一个流值 λ 的最小代价流 f，则 $N(f)$ 中没有负回路，故可求出 $N(f)$ 中最短 s-t 路 P(关于权 w^0). 令

$$\delta=\min\{c^0(P),\lambda-\text{val} f\},$$

并作 f 关于 P 与 δ 的修改流 $\hat{f}_\delta$，则 $\hat{f}_\delta$ 是流值$(\text{val} f+\delta)$ 的最小代价流. 再以 $\hat{f}_\delta$ 代 f，重复迭代直至求出流值 λ 的最小代价流.

算法 8.4 最小代价流的迭加算法

设 $N=(V,E,c,w,s,t)$，没有权 w 的负回路.

第 1 步 求流值不超过 λ 的最小代价流 f(例如零流).

第 2 步 若 $\text{val} f=\lambda$，则 f 是流值 λ 的最小代价流. 否则作具代价的伴随网络 $N(f)$.

第 3 步 若 $N(f)$ 中没有 s-t 路，则 N 中没有流值 λ 的流，此时，f 是最小代价最大流.

第 4 步 设 P 是最短 s-t 路，令

$$\delta=\min\{c^0(P),\lambda-\text{val} f\},$$

作 f 关于 P 和 δ 的修改流 $\hat{f}_\delta$. 令

$$f=\hat{f}_\delta,\ \text{val} f=\text{val} f+\delta,$$

转第 2 步.

例 8.11 求图 8.8(a) 的流值 4 的最小代价流，设初始流为零流. 其解的过程如图 8.11.

N 与 f	N(f)	P
val $f=0$		$P=sv_2t$ $c^*(p)=3$ $\lambda-\mathrm{val}\,f=4$ $\delta=3$
val f=0+3=3		$P=sv_1v_2t$ $c^*(p)=1$ $\lambda-\mathrm{val}\,f=1$ $\delta=1$
val f=3+1=4	$\because$ val $f=4$ $\therefore$ f是流值4的最小代价流	

图 8.11 迭加算法

网络 N 中箭杆旁的数字是(f,c,w),具代价的伴随网络 $N(f)$ 中箭杆旁的数字是(c^0,w^0)

§8.5 应用——开关网络

定义 8.13 一个二端元件称为**开关**,如果恰有二个状态:接通状态和断开状态.在接通状态开关两端电流流通,在断开状态开关两端电流不流通.开关可用一条边表示,边的权为 1 表接通状态,权为 0 表断开状态.

由开关组成的无向网络称为**开关网络**.换言之,开关网络是一个无向连通图,且每条边对应一个布尔变量为该边的权.于是,开关网络的边可看作是一个布尔变量.

我们用"+"表布尔加,用"·"表布尔乘,用"$\bar{x}$"表 x 的补.

定义 8.14 设 v_i,v_j 是开关网络中两个顶点,P_{ij} 是一条 v_i-v_j 路,则表示路 P_{ij} 的边的布尔变量的乘积称为**路积**.为方便,我们把路的边集或路积就作为路.

用 f_{ij} 表示 v_i 与 v_j 间所有路的路积的布尔和,称为顶点 v_i 与 v_j 的**开关函数**.

例 8.12 图 8.12(a)是由 5 个开关组成的开关网络,其模拟图如图 8.12(b).v_i 与 v_j 间有 4 条路:$\{x_1,x_2\}$,$\{x_1,x_3,x_5\}$,$\{x_4,x_5\}$ 与 $\{x_2,x_3,x_4\}$.于是顶点 v_i 与 v_j 的开关函数是

$$F=\begin{bmatrix} 1 & x_1+x_2 & (x_1+x_2)\bar{x}_3+x_3 & (x_1+x_2)\bar{x}_3x_4+x_3x_4 \\ x_1+x_2 & 1 & (x_1+x_2)x_3+\bar{x}_3 & (x_1+x_2)x_3x_4+\bar{x}_3x_4 \\ (x_1+x_2)\bar{x}_3+x_3 & (x_1+x_2)x_3+x_3 & 1 & x_4 \\ (x_1+x_2)\bar{x}_3x_4+x_3x_4 & (x_1+x_2)x_3x_4+\bar{x}_3x_4 & x_4 & 1 \end{bmatrix} \tag{8.28}$$

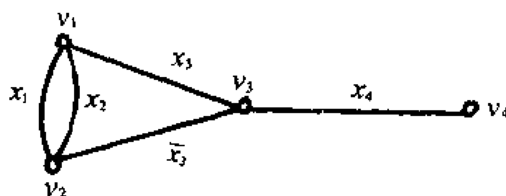


图 8.14

注意到 z_{ij} 等于顶点 v_i 与 v_j 的不超过一个开关的开关函数，f_{ij} 等于顶点 v_i 与 v_j 的不超过 $(n-1)$ 个开关的开关函数，故容易得到下面的定理（习题 8.36）.

定理 8.15 设 Z 与 F 分别是有 n 个顶点的开关网络的连接矩阵与传输矩阵，则

$$F = Z^{n-1}. \tag{8.29}$$

定义 8.17 若开关网络的每个开关都能独立地接通或断开，则称为**单接触网络**.

由定义，单接触网络的开关变量都是各不相同的，因此必是最小化的.

图 8.13(b) 的网络是单接触网络，图 8.13(a) 与图 8.14 的网络都不是单接触网络.

定义 8.18 设开关网络有 m 条边，顶点 u 与 v 间有 l 条路，令

$$w_{ij}=\begin{cases}0, & \text{边 } j \text{ 不在第 } i \text{ 条路上},\\ 1, & \text{边 } j \text{ 在第 } i \text{ 条路上},\end{cases} \tag{8.30}$$

则 $W=(w_{ij})_{l\times m}$ 称为顶点 u 与 v 的**路矩阵**.

例 8.15 单接触网络的路矩阵与开关函数有着对应关系：开关函数的项对应路矩阵的行；在开关函数的第 i 项中，

$$w_{ij}=\begin{cases}1, & \text{变量 } x_j \text{ 出现},\\ 0, & \text{变量 } x_j \text{ 不出现}.\end{cases}$$

例如，对图 8.12(b) 的 v_i 与 v_j，从开关函数式(8.23) 可由上面的对应关系直接写出路矩阵

$$W=\begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix}$$
$$W=\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

定义 8.19 设 v_i 与 v_j 是开关网络的两个顶点，若附加一条边 x_0 连接 v_i 与 v_j 使网络图是不可分图，则称该网络为**对顶点 v_i 与 v_j 是不可分的**，有时也简称为**不可分二端图**.

图 8.13(b) 是不可分二端图.

不可分二端图的一个重要性质是，任一圈能表为两条 v_i-v_j 路的对称差.

定理 8.16 设开关网络 N 对顶点 v_i 与 v_j 不可分，则网络 N 的任一圈能表为两条 v_i-v_j

路的对称差.

证 设 C 是一个圈.若 C 过 v_i 与 v_j,则 C 由两条 v_i-v_j 路 $P_{ij}^{(1)}$ 与 $P_{ij}^{(2)}$ 组成,故

$$C = P_{ij}^{(1)} \oplus P_{ij}^{(2)}, \tag{8.31}$$

若 C 过 v_i 不过 v_j,设 v_k 是 C 上异于 v_i 的顶点,因为 v_i 不是分离点,故有 v_k-v_j 路不过 v_i.今从 v_k 出发沿 P 旅行,与 C 的最后一个公共顶点是 v_r,记 P 的 v_r-v_j 段为 P_{rj}.顶点 v_r 分 C 为两条 v_i-v_r 路 $P_{ir}^{(1)}$ 与 $P_{ir}^{(2)}$,令

$$P_{ij}^{(1)} = P_{ir}^{(1)} \cup P_{rj},\ P_{ij}^{(2)} = P_{ir}^{(2)} \cup P_{rj},$$

则式(8.31)成立.若 C 不过 v_i 与 v_j,设 v_k 与 v_l 是 C 上相异两顶点,则有 v_i-v_l 路 P_1 不过 v_j,也有 v_k-v_j 路 P_2 不过 v_i.从 v_i 出发沿 P_1 与 C 的第一个公共顶点是 v_s,记 P_1 的 v_i-v_s 段为 P_{is}.从 v_k 出发沿 P_2 与 C 的最后一个交点是 v_r,记 P_2 的 v_r-v_j 段为 P_{rj}.顶点 v_s 与 v_r 分 C 为两条 v_s-v_r 路 $P_{sr}^{(1)}$ 与 $P_{sr}^{(2)}$,令

$$P_{ij}^{(1)} = P_{is} \cup P_{sr}^{(1)} \cup P_{rj}, \quad P_{ij}^{(2)} = P_{is} \cup P_{sr}^{(2)} \cup P_{rj},$$

则式(8.31)成立.综上得证. **证毕.**

由定理 8.16,不可分二端图的圈向量能表为路矩阵两个行向量的对称差.换言之,网络的所有圈能由路矩阵行向量的对称差产生.

对一个 v_i 与 v_j 的不可分二端网络,在所有 v_i-v_j 路中有多少条路是独立的?我们有下面的定理.

定理 8.17 设开关网络 N 有 n 个顶点 m 条边,且对顶点 v_i 与 v_j 是不可分的,则 v_i 与 v_j 的路矩阵 W 的秩是

$$r(W) = m - n + 2. \tag{8.32}$$

证 用边 x_0 连接 v_i 与 v_j 构成新的网络 N_0,则 N 的所有 v_i-v_j 路加上边 x_0 构成 N_0 的含 x_0 的所有圈,设圈矩阵为 C,则

$$C = (W\ \ 1)_{l\times(m+1)}, \tag{8.33}$$

其中 1 是 l 维列向量,每个元素都是 1.由定理 8.16,N 的所有圈由 W 的任二行的对称差得出,因此,N_0 的所有圈能由 C 的行的对称差得出.所以 C 的秩等于 N_0 的圈秩,即

$$r(C) = (m+1) - n + 1 = m - n + 2,$$

又 $r(W) = r(C)$,故式(8.32)成立. **证毕.**

定义 8.20 路矩阵的任意$(m-n+2)$个独立行构成的子矩阵称为**基本路矩阵**,记为 W_f.基本路矩阵对应$(m-n+2)$条独立的路,称为**基本路**.

现在讨论当给定开关函数 f_{ij} 时,如何实现这个网络,即设计一个开关网络 N 具有开关函数 f_{ij}.

由路矩阵与开关函数的对应关系,容易从给定开关函数 f_{ij} 写出路矩阵 W.引入边 x_0 从网络 N 得网络 N_0,并由式(8.33)得矩阵 C.对 C 经行变换得 N_0 的基本圈矩阵 B_f.然后由定理 3.27 写出基本余圈矩阵 Q_f.因为基本余圈矩阵与基本关联矩阵都是余圈空间的基矩阵,所以从 Q_f 经行变换可得基本关联矩阵 A_f.再从 A_f 得关联矩阵 A.最后去掉边 x_0 的列即得 N 的关联矩阵.由此得到的网络正是开关函数 f_{ij} 的实现.

算法 8.5 开关函数的单接触网络实现的算法

设 f_{ij} 是给定开关函数.

第 1 步　写出路矩阵 W.

第 2 步　引入边 x_0，由式(8.33) 构造 N_0 的含 x_0 的圈矩阵 C.

第 3 步　对 C 用对称差求基本圈矩阵

$$B_f = (I_{m-n+2}, B_{f,2}).$$

第 4 步　由定理 3.27 写出基本余圈矩阵

$$Q_f = (B_{f,2}^T, I_{n-1}).$$

第 5 步　用初等行变换化 Q_f，使每列至多 2 个 1 而得基本关联矩阵 A_f.

第 6 步　在 A_f 补充 1 行，使所得矩阵每列正好 2 个 1 而得 N_0 的关联矩阵 A.

第 7 步　在 A 中去掉 x_0 的列即得 N 的关联矩阵，此即开关函数 f_{ij} 的单接触网络实现.

例 8.16　给定开关函数如式(8.24)，求其实现.

首先由式(8.24) 写出路矩阵

$$W = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array},$$

再由式(8.33) 写出矩阵 C,

$$C = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_0 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array},$$

在 C 中已经有一个单位阵，故

$$B_f = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_0 \end{matrix} \\ \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right) \end{array} = (I_{m-n+2}, B_{f,2}),$$

由此写出 Q_f,

$$Q_f = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_0 \end{matrix} \\ \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right] \end{array},$$

把 Q_f 的第 1 行加到第 2 行，使每列至多 2 个 1 而得到基本关联矩阵 A_f,

$$A_f = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_0 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array},$$

附加 1 行，使 A_f 每列正好 2 个 1 得关联矩阵 A,

$$A = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_0 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array},$$

去掉 x_0 的列得 N 的关联矩阵

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

这正是图 8.13(b) 的开关网络.

§8.6 应用——网络计划技术

网络计划技术又称统筹方法，是在国防、工业、经济、管理、基本建设等方面组织实施工作计划的一种数学方法. 大量实践说明，网络计划技术能使计划工作做到统筹兼顾，全面安排，抓住关键. 对于生产技术复杂或者跨部门跨行业的大型工程计划，或大型科研计划，网络计划技术的优点尤为突出.

定义 8.21 一项工程由若干相互独立的活动组成，每个活动称为**工序**. 相邻两个工序在时间上的分界点称为**事项**，它表示紧前工序的结束和紧后工序的开始. 我们用顶点表示事项，用有向边表示工序. 边的起点表示工序的开始，称为该工序的**开工事项**；终点表示工序的结束，称为该工序的**完工事项**. 边的**权**是工序的完成时间. 用一个顶点表示整个工程计划的开始，该顶点称为**起点事项**. 用另一个顶点表示整个工程计划的结束，该顶点称为**终点事项**. 这样就得到一个网络 $N=(V,\vec{E},w,s,t)$，其中 $G=(V,E)$ 是赋权有向图，每条边 $e\in E$ 有一个权 $w(e)$ 即该工序的完成时间，s 是起点事项，t 是终点事项，称为**工程网络**或**统筹图**.

设 $e=(i,j)$，则 e 的权 $w(e)$ 也记为 w_{ij}.

工序表示一项活动，需消耗资源，需一定时间. 事项是时间的瞬间，表示工序的开始或结束，不消耗资源，也不需时间.

一个工程网络应注意下列五方面.

(1) 工序是具有相对独立性的活动，不同的工序用不同的边表示，反过来，不同的边也表示不同的工序，并且不能有平行边，否则造成工序相对独立性的混乱.

(2) 箭头方向表示工序的顺序关系. 若工序 A 是工序 B 的紧前工序，则表示工序 A 的边的完工事项，就是表示工序 B 的边的开工事项. 箭头方向从左到右(虚箭头可从上到下或从下到上). 起点事项在工程网络最左边，终点事项在工程网络最右边.

(3) 不能有有向圈，否则造成工序顺序关系的混乱.

(4) 为方便计算，工程网络的每个顶点需用数字编号，即每个顶点 v 对应一个正整数 $l(v)$，并要求对每条边 $(u,v)\in E$，都有 $l(u)<l(v)$. 由习题 1.40，这样的标号是能实现的. 起点事项标号最小，设为 1. 终点事项标号最大，正好是网络的顶点数，设为 n. 为方便可直接用顶点标号称呼顶点或事项.

(5) 为确切表示工序的衔接关系，需要引进虚工序. **虚工序**即**权**为零的工序，用虚箭

头表示.虚工序不消耗资源,也不需时间,在工程网络中的作用仅仅表明工序的衔接关系.

（Ⅰ）**平行作业**.工序 C 需在工序 A 与 B 都完工之后才能开工,则工序 A 与 B 称为平行作业,为避免平行边,应引入虚工序,如图 8.15(a).

（Ⅱ）**交叉作业**.假定工序 B 的紧前工序是 A.如果工序 B 只需在工序 A 部分完工之后即可开工,则可把工序 A 分解为子工序 A_1 与 A_2,把工序 B 分解为子工序 B_1 与 B_2,使 A 的子工序 A_2 与 B 的子工序 B_1 交叉作业.如图 8.15(b) 表示 A_1 完工后 A_2 与 B_1 即可开工,当 A_2 与 B_1 都完工后 B_2 才能开工.

（Ⅲ）**一个起点事项**.当整个工程的开始多于一个工序时,可引入一个起点事项和相应的虚工序,如图 8.15(c).

（Ⅳ）**一个终点事项**.当整个工程的结束多于一个工序时,可引入一个终点事项和相应的虚工序,如图 8.15(d).

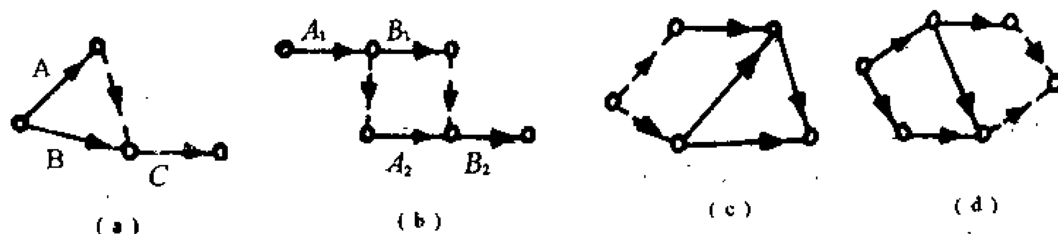


图 8.15　虚工序的应用

(a) 平行作业　(b) 交叉作业　(c) 一个起点事项　(d) 一个终点事项

例 8.17　已知工序,工序时间与紧前工序如表 8.2,试画出工程网络图.

表 8.2　工序表

工序	A	B	C	D	E	F	G	H	I	J	K
工序时间	2	4	7	3	10	2	4	3	2	3	2
紧前工序	/	A	A	A	B	C,D,K	E,F	D	H	G,I	B

第 1 步　给工序分级,没有紧前工序的作为 1 级;删除 1 级工序后没有紧前工序的作为 2 级;依次类推.由表 8.2,工序分类如表 8.3.

表 8.3　工序分级表

级	1	2	3	4	5	6
工序	A	B,C,D	E,H,K	F,I	G	J

第 2 步　按工序级别从左到右排列,并按工序的顺序画出网络图,如图 8.16.

第 3 步　对事项编号.起点事项的入度为零,标 1;去掉已经标号的顶点,对入度为零的事项依次标号;继续之,直至终点事项得到最大标号为止.如图 8.16.

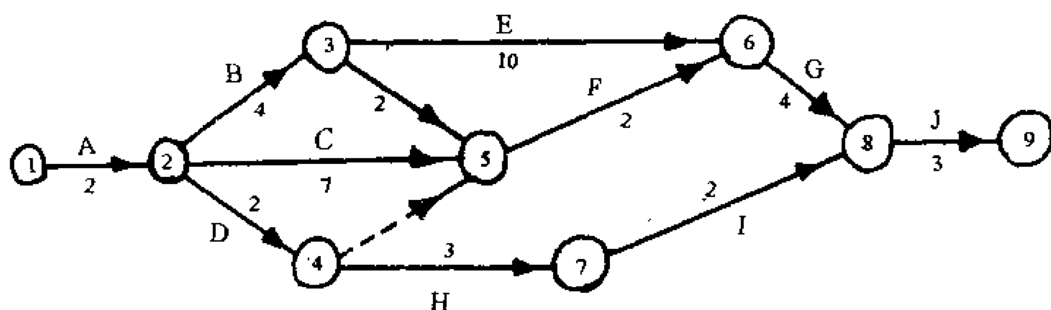


图 8.16 工程网络图

定义 8.22 事项k的**最早日期**$t_E(k)$，表示在整个工程开始后，在以k为完工事项的所有工序如期完工的条件下，事项k的最早执行时间. 整个工程从开始到结束的时间称为**总工期**，记为T_E.

由定义，$t_E(k)$是从顶点1沿所有有向路到顶点k，事项k的最早执行时间，即从顶点1到顶点k的最长路的权，故有递推公式

$$\begin{cases} t_E(1) = 0, \\ t_E(k) = \max\ \{t_E(i) + w_{ik} \mid (i,k) \in E\},\ k > 1. \end{cases} \tag{8.34}$$

设工程网络有n个事项，则终点事项为n. 由定义，$t_E(n)$是顶点1到顶点n的最长路的权，即整个工程从开始到结束的时间，故

$$T_E = t_E(n). \tag{8.35}$$

定义 8.23 事项k的**最迟日期**$t_L(k)$，表示在不误总工期的条件下，以k为开工事项的所有工序如期开工，事项k的最迟执行时间.

因为终点事项n表示整个工程的结束，所以在不误总工期的条件下，事项n的最迟执行时间与最早执行时间相等，即

$$t_L(n) = T_E.$$

对于事项$k(k<n)$，由定义，$t_E(k)$是从顶点k沿所有的路到顶点n，事项k的最迟执行时间，即从顶点k沿最长路到n，k的最迟执行时间，故有

$$\begin{aligned} T_E - t_L(k) &= k\text{ 到 }n\text{ 的最长路的权} \\ &= \max\ \{w_{kj} + j\text{ 到 }n\text{ 的最长路的权} \mid (k,j) \in E\} \\ &= \max\ \{T_E - t_L(j) + w_{kj} \mid (k,j) \in E\}, \\ t_L(k) &= T_E - \max\ \{T_E - t_L(j) + w_{kj} \mid (k,j) \in E\} \\ &= \min\ \{t_L(j) - w_{kj} \mid (k,j) \in E\}, \end{aligned}$$

于是得递推公式

$$\begin{cases} t_L(n) = T_E, \\ t_L(k) = \min\ \{t_L(j) - w_{kj} \mid (k,j) \in E\}. \end{cases} \tag{8.36}$$

例 8.18 求图 8.16 的工程网络中事项的最早日期与最迟日期.

由式(8.34)，计算事项的最早日期，依事项编号由小到大的顺序递推. 例如，

$t_E(1) = 0$

$t_E(2) = t_E(1) + w_{12} = 0 + 2 = 2$

$t_E(3) = t_E(2) + w_{23} = 2 + 4 = 6$

$t_E(4) = t_E(2) + w_{24} = 2 + 3 = 5$

$t_E(5) = \max\{t_E(2) + w_{25}, t_E(3) + w_{35}, t_E(4) + w_{45}\}$

$= \max\{2 + 7, 6 + 2, 5 + 0\} = 9$

等等,全部结果如表 8.4.

由式(8.36),计算事项的最迟日期,依事项编号由大到小的顺序递推.例如,

$t_L(9) = 23$

$t_L(8) = t_L(9) - w_{89} = 23 - 3 = 20$

$t_L(7) = t_L(8) - w_{78} = 20 - 2 = 18$

$t_L(6) = t_L(8) - w_{68} = 20 - 4 = 16$

$t_L(5) = t_L(6) - w_{56} = 16 - 2 = 14$

等等,全部结果如表 8.4.

表 8.4　**事项的时间参数**

事项 k	1	2	3	4	5	6	7	8	9
$t_E(k)$	0	2	6	5	9	16	8	20	23
$t_L(k)$	0	2	6	14	14	16	18	20	23

上面讨论了工程网络关于事项的时间参数,这是网络计算的基础.现在讨论工序的时间参数.

(1) 工序(i,j)的**最早开始时间** $ES(i,j)$.

∵　$t_E(i)$ = 以 i 为完工事项的所有工序如期完工,事项 i 的最早执行时间

= 以 i 为开工事项的所有工序的最早开始时间,

∴　$ES(i,j) = t_E(i)$.　(8.37)

(2) 工序(i,j)的**最早结束时间** $EF(i,j)$.

$$EF(i,j) = ES(i,j) + w_{ij} = t_E(i) + w_{ij}. \quad (8.38)$$

(3) 工序(i,j)的**最迟结束时间** $LF(i,j)$.

∵　$t_L(j)$ = 以 j 为开工事项的所有工序如期开工,事项 j 的最迟执行时间

= 以 j 为完工事项的所有工序的最迟结束时间,

∴　$LF(i,j) = t_L(j)$.　(8.39)

(4) 工序(i,j)的**最迟开始时间** $LS(i,j)$.

$$LS(i,j) = LF(i,j) - w_{ij} = t_L(j) - w_{ij}. \quad (8.40)$$

例 8.19　求图 8.16 的工程网络中工序的时间参数.

由式(8.37) ~ (8.40),利用表 8.4 的结果可得工序的 4 个时间参数如表 8.5.

表 8.5　工序的时间参数

工序	w_{ij}	$ES(i,j)$	$EF(i,j)$	$LS(i,j)$	$LF(i,j)$
$A=(1,2)$	2	0	2	0	2
$B=(2,3)$	4	2	6	2	6
$C=(2,5)$	7	2	9	7	14
$D=(2,4)$	3	2	5	11	14
$E=(3,6)$	10	6	16	6	16
$F=(5,6)$	2	9	11	14	16
$G=(6,8)$	4	16	20	16	20
$H=(4,7)$	3	5	8	15	18
$I=(7,8)$	2	8	10	18	20
$J=(8,9)$	3	20	23	20	23
$K=(3,5)$	2	6	8	12	14

定义 8.24　工序(i,j)的**总时差** $R(i,j)$ 是在不误总工期的条件下，工序(i,j)的开工时间可以推迟的最长时间. 工序(i,j)的**局部时差** $r(i,j)$ 是在不误所有紧后工序的最早开工日期的条件下，工序(i,j)的开工时间可以推迟的最长时间. 总时差与局部时差如示意图 8.17.

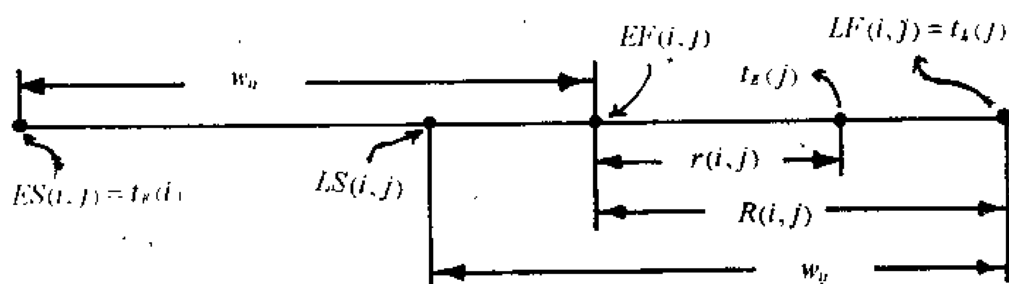


图 8.17　总时差与局部时差关系示意图

由定义，工序(i,j)的最早开始时间加上总时差正好等于工序(i,j)的最迟开始时间，即

$$ES(i,j)+R(i,j)=LS(i,j).$$

于是

$$\begin{aligned}R(i,j)&=LS(i,j)-ES(i,j)\\&=LF(i,j)-EF(i,j)\\&=t_L(j)-t_E(i)-w_{ij}.\end{aligned}\tag{8.41}$$

由定义，工序(i,j)的最早结束时间加上局部时差正好等于以 j 为开工事项的最早执行时间，即

$$EF(i,j)+r(i,j)=t_E(j).$$

于是

$$r(i,j) = t_E(j) - EF(i,j)$$
$$= t_E(j) - t_E(i) - w_{ij}. \tag{8.42}$$

例 8.20　求图 8.16 的工程网络中工序的总时差与局部时差.

由式(8.41)与(8.42),计算结果如表 8.6.

表 8.6　总时差与局部时差

工序	w_{ij}	$R(i,j)$	$r(i,j)$
$A=(1,2)$	2	0	0
$B=(2,3)$	4	0	0
$C=(2,5)$	7	5	0
$D=(2,4)$	3	9	0
$E=(3,6)$	10	0	0
$F=(5,6)$	2	5	5
$G=(6,8)$	4	0	0
$H=(4,7)$	3	10	0
$I=(7,8)$	2	10	10
$J=(8,9)$	3	0	0

综合以上计算，可把结果画在工程网络图中，一目了然，便于观察处理．工程网络图如图 8.18(a)，图中粗线表示关键路，箭杆旁的符号的意义如图 8.18(b)，顶点旁的符号的意义如图 8.18(c)．

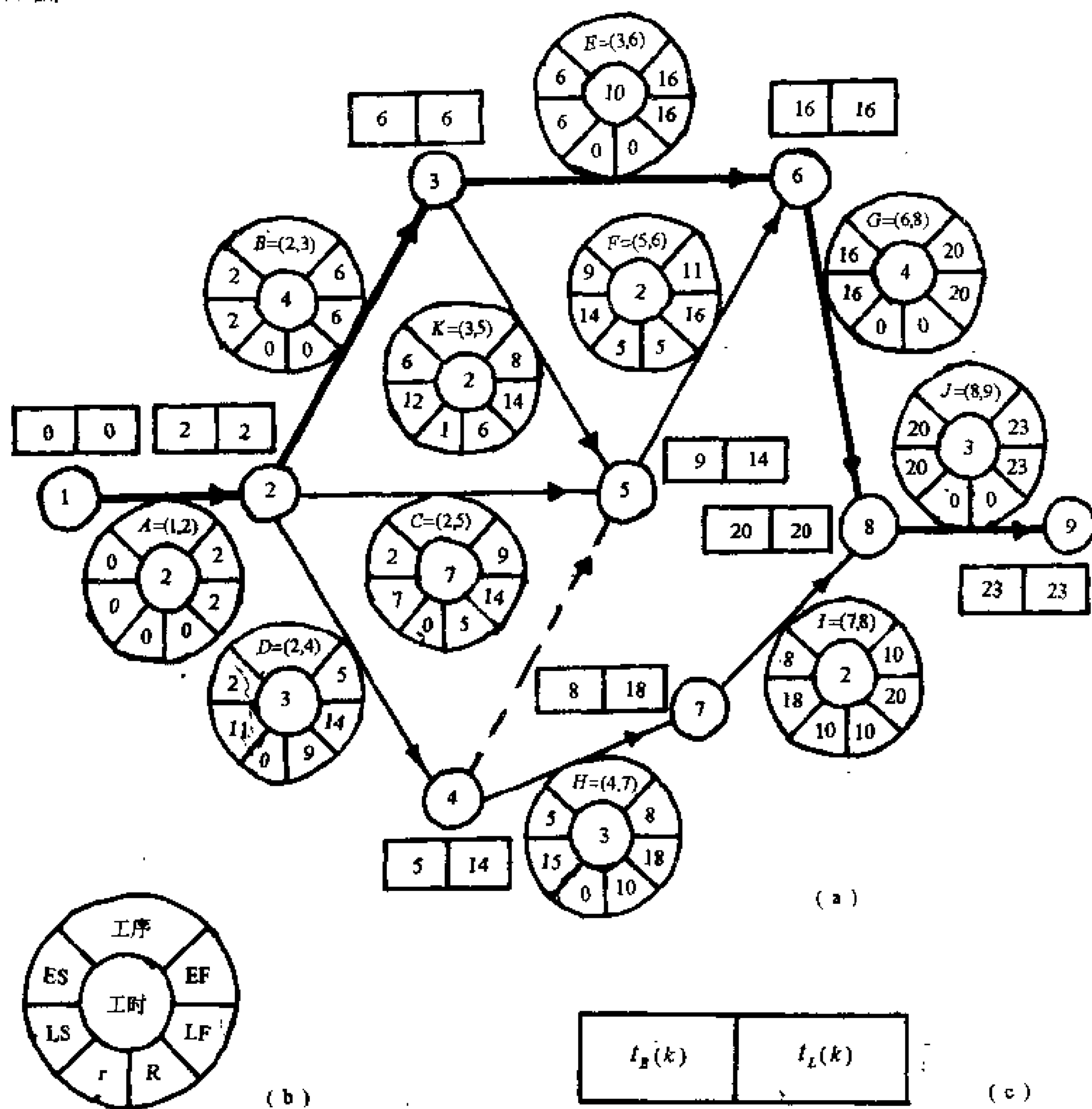


图 8.18　工程网络计算

若 k 是关键事项，由引理 8.18，事项 k 的最早日期 $t_E(k)$ 与最迟日期 $t_L(k)$ 相等，即事项 k 的执行时间没有任何机动，必须按计划如期执行，否则必然延长总工期，推迟整个工程的结束时间．若 (i,j) 是关键工序，由引理 8.19，总时差为零，即工序 (i,j) 的开工时间不允许推迟，否则将延长总工期．由此即知，由关键事项与关键工序组成的关键路，的确是整个工程的关键．关键工序延期，则关键路的长增加，整个工程就延期；关键工序提前，则关键路的长减少，整个工程就提前．因此，在整个工程的统筹安排中，我们的策略应当是向关键工序要时间．其方法可考虑如下：

(1) 提高关键工序的效率，例如改进组织，更新设备，技术革新，增加投入，适当加班等；

(2) 对关键工序用交叉作业，平行作业等方法以缩短工序的工期；

(3) 从非关键工序获得支援.

设 (i,j) 是非关键工序，由引理 8.19，$R(i,j) > 0$，即在不误总工期的条件下，工序 (i,j) 的开工时间可以推迟 $R(i,j)$，换言之，工序 (i,j) 有 $R(i,j)$ 的富余时间. 因此，为使整个工程提前，我们的策略应当是**向非关键工序要资源**，充分挖潜以支援关键工序. 其方法可考虑如下：

(1) 非关键工序如期完工，把富余时间的剩余资源支援关键工序；

(2) 非关键工序在不超过富余时间的范围内延期完工，以节约资源支援关键工序.

算法 8.6 网络计划技术的算法

第 1 步 确定工序，工序时间，各工序的紧前工序.

第 2 步 画出工程网络图：

(1) 给工序分级并列出工序分级表. 分级的方法是：没有紧前工序的作为第 1 级；删除第 1 级工序后没有紧前工序的作为第 2 级；余类推.

(2) 从左至右画出网络图，起点事项在最左边，终点事项在最右边.

(3) 给事项编号，起点事项的入度为零，编号为 1. 去掉已编号事项，对入度为零的事项依次编号. 最后，终点事项编号为最大. 所得编号必然满足对任意的边 (i,j)，i 的编号小于 j 的编号.

第 3 步 计算事项时间参数：

(1) 从起点事项至终点事项，依从小到大的编号顺序，由式(8.34)递推求出事项的最早日期.

(2) 从终点事项至起点事项，依从大到小的编号顺序，由式(8.36)递推求出事项的最迟日期.

第 4 步 计算工序时间参数：

(1) 由式(8.37)计算 $ES(i,j)$.

(2) 由式(8.38)计算 $EF(i,j)$.

(3) 由式(8.39)计算 $LF(i,j)$.

(4) 由式(8.40)计算 $LS(i,j)$.

(5) 由式(8.41)计算 $R(i,j)$.

(6) 由式(8.42)计算 $r(i,j)$.

第 5 步 找出关键事项，关键工序，关键路.

第 6 步 将网络的各参数画在工程网络图中(如图 8.18 的形式).

第 7 步 根据"向关键工序要时间，向非关键工序要资源"的原则调整网络计划，重新绘制网络图即返回第 1 步.

第 8 步 当所有工序或尽可能多的工序都成为关键工序时，所得网络计划就是优化的. 这里的优化是指已经使所有工序或尽可能多的工序在不误总工期的条件下没有富余时间，换言之，已经尽可能提高工效和挖掘潜力使总工期不能再缩短.

当然，在制订任何计划时应该留有余地，否则计划的实施也许会遇到困难或甚至落

空.留有余地的原则一般体现在工时的确定，即在确定工序的完成时间时必须留有余地.余地的限度可以参考过去有关工程的数据和有关规定，并找有关专家和工程人员进行科学评估.

§8.7 应用——前导网络

在网络计划技术中前面所讨论的工程网络，是用边表示工序，用顶点表示事项，这样的网络称为**事项网络**.在事项网络中，一个工序的开工，必须等到所有紧前工序都完工之后.这种由结束到开始的顺序关系称为**结束到开始**，记为 FS.工序之间的关系还有另外三种：**结束到结束**(FF)，**开始到开始**(SS)，**开始到结束**(SF).这四种顺序关系统称为**前导关系**.为更好地反映工序间的这些关系，采用工序网络是方便的.**工序网络**是用顶点表示工序，边的方向表示工序的各种顺序关系的工程网络.

例 8.22 举例说明工序的前导关系，如图 8.19.

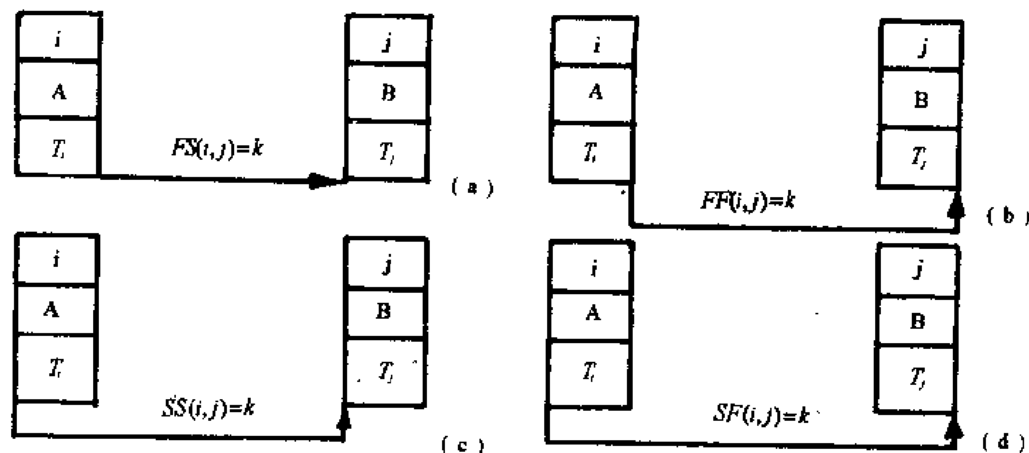


图 8.19 工序的四种前导关系

在工序网络中，表示工序的顶点用方框，每个方框分为 3 格，中间一格是工序，上边一格是工序的编号，下边一格是工序的完成时间.

在图 8.19 的四种前导关系中，工序 A 与 B 的编号分别是 i 与 j，工时分别是 T_i 与 T_j，并称工序 A 是工序 B 的**前导工序**，B 是 A 的**后继工序**.

在图 8.19 中，方框左边竖线表示工序的开始，右边竖线表示工序的结束.$FS(i,j)=k$ 表示工序 i 结束 k 天(假定时间以天计)后工序 j 才能开始，箭头是从方框右边(结束)指向方框左边(开始)，如图 8.19(a)；余类推，如图 8.19(b) ~ (c).k 称为**滞后量**.

我们称工序 i 是工序 j 的**紧前工序**，即工序 i 是工序 j 的 FS 滞后量为零的前导工序.

定义 8.26 能表示工序间的前导关系的工序网络称为**前导网络**.前导网络只有一个开始工序 S，它在网络图的最左边；只有一个结束工序 T，它在网络图的最右边.开始工序与结束工序表示整个工程的开始与结束，其工时都是零.

例 8.23 按表 8.2 画出工序网络.

引入开始工序 S 与结束工序 T，画出工序网络，并给各工序编号，使前导工序的编号小于后继工序的编号，如图 8.20.

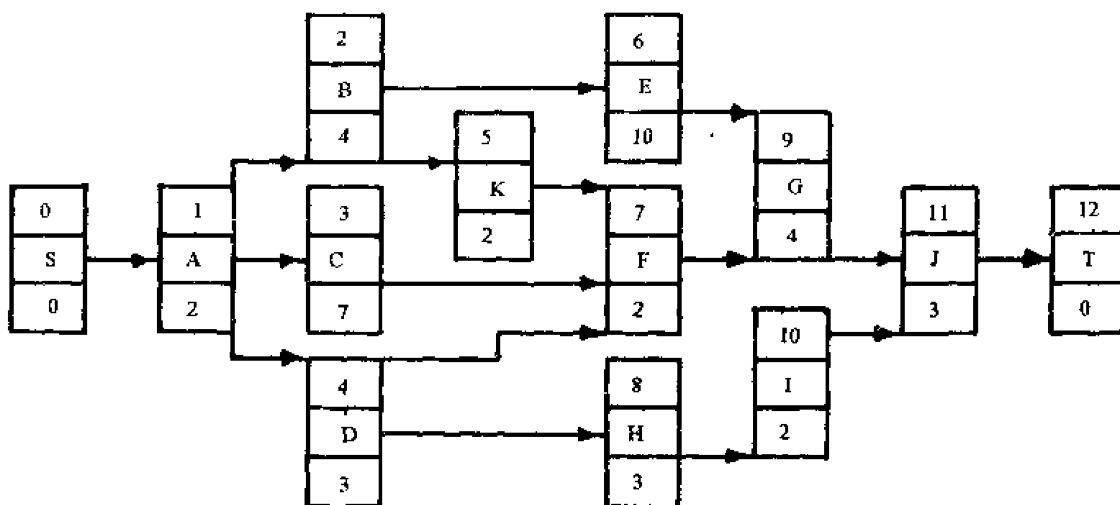


图 8.20 工序网络

在图 8.20 中，由于 FS 的滞后量为零，因此没有在箭杆上标出.

算法 8.7 工序网络时间参数计算法

第 1 步 **工序 i 的最早开始时间 ES_i，最早结束时间 EF_i.**

每个工序 i，必须在所有前导工序都完工之后才能开工，换言之，ES_i 是从开始工序（用顶点 0 表示）沿所有路到工序 i，工序 i 的最早开始时间，即从顶点 0 到顶点 i 的开始时的最长时间. 于是 EF_i 是从顶点 0 到顶点 i 的结束时的最长时间. 故有如下递推公式（从顶点 0 递推到 n，n 表示结束工序）：

$$\begin{cases} ES_0 = EF_0 = 0, \\ ES_i = \max \{EF_k | k \in \Gamma^-(i)\}, \ i > 0, \\ EF_i = ES_i + T_i, \qquad\qquad i > 0, \end{cases} \tag{8.45}$$

其中 T_i 是工序 i 的工时.

第 2 步 **总工期 T_E.**

因结束工序 T 的编号为 n，故 EF_n 表示从顶点 0 到顶点 n 的结束时的最长期间，即整个工程的完成时间，故

$$T_E = EF_n. \tag{8.46}$$

第 3 步 **相邻工序 i 与 j 的独立时差 $IF(i,j)$.**

设工序 i 的后继工序之一是工序 j，则在不误工序 j 最早开始时间的条件下，工序 i 可以延期完工的最长时间称为独立时差，由此，

$$EF_i + IF(i,j) = ES_j,$$

$$IF(i,j) = ES_j - EF_i. \tag{8.47}$$

第 4 步 **工序 i 的局部时差 r_i.**

在不误工序 i 的所有后继工序的最早开始时间的条件下，工序 i 可以延期完工的最长时间称为局部时差，由此，

$$\begin{aligned} r_i &= \min\{IF(i,j) \mid j \in \Gamma^+(i)\} \\ &= \min\{ES_j - EF_i \mid j \in \Gamma^+(i)\}. \end{aligned} \tag{8.48}$$

第 5 步　**工序 i 的总时差 R_i.**

在不误总工期的条件下，工序 i 可以延期完工的最长时间称为总时差. 对于结束工序 T，不允许延后完工，故 $R_n = 0$. 对任一工序 $i < n$，设工序 j 是 i 的一个后继工序，则

$$IF(i,j) + R_j$$

表示从顶点 i 经过顶点 j 到顶点 n，在不误总工期的条件下，工序 i 可以延期完工的最长时间，故有

$$R_i = \min\{IF(i,j) + R_j \mid j \in \Gamma^+(i)\}.$$

再由式(8.47)得递推公式(从顶点 n 递推到 0)：

$$\begin{cases} R_n = 0, \\ R_i = \min\{ES_j - EF_i + R_j \mid j \in \Gamma^+(i)\},\ i < n. \end{cases} \tag{8.49}$$

第 6 步　**工序 i 的最迟开始时间 LS_i，最迟结束时间 LF_i.**

在不误总工期的条件下，工序 i 的最迟开始时间，即工序 i 的最早开始时间与总时差的和，故有

$$LS_i = ES_i + R_i, \tag{8.50}$$

$$\begin{aligned} LF_i &= LS_i + T_i \\ &= EF_i + R_i. \end{aligned} \tag{8.51}$$

第 7 步　**确定关键路.**

总时差为零的工序是关键工序，因此关键路是从开始工序沿总时差为零的顶点到结束工序的路，即从顶点 0 到顶点 n 的最长路.

事项网络与工序网络时间参数计算的比较

(Ⅰ) 事项网络的基础参数是事项的最早日期与最迟日期，其它时间参数都可由它们求出. 工序网络的基础参数是工序的最早开始时间与最早结束时间，其它时间参数都可由它们求出.

(Ⅱ) 事项网络有 2 个事项时间参数，6 个工序时间参数，加上总工期，共 9 个时间参数. 工序网络有 6 个工序时间参数，加上独立时差和总工期，共 8 个时间参数.

(Ⅲ) 从图的观点看，事项网络与工序网络的 2 个基础时间参数都是关于顶点的.

(Ⅳ) 事项网络与工序网络相一致的时间参数是 6 个工序时间参数和总工期. 不一致的时间参数，在事项网络中是 2 个事项参数，在工序网络中是独立时差.

例 8.24　计算图 8.20 的工序网络的时间参数并找出关键路. 计算结果如图 8.21(c). 顶点处符号的意义如图 8.21(a)，箭杆旁的数字为独立时差，如图 8.21(b).

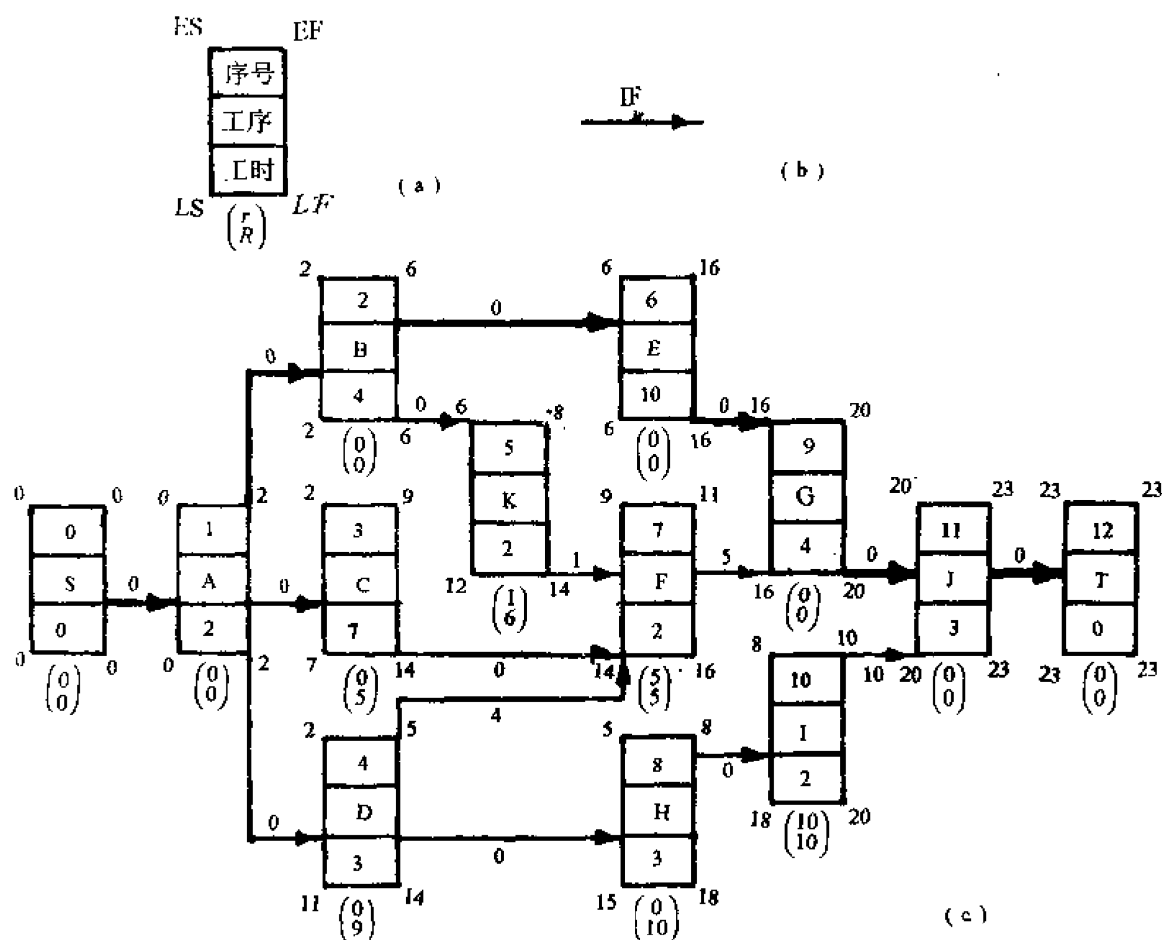


图 8.21　工序网络的计算

算法 8.8　前导网络时间参数计算法

第 1 步　工序的最早开始时间和最早结束时间.

1. 开始到开始关系，由图 8.19(c)，

$$\begin{cases} ES_j = ES_i + SS(i,j), \\ EF_j = ES_j + T_j. \end{cases} \tag{8.52}$$

2. 结束到开始关系，如图 8.19(a)，

$$\begin{cases} ES_j = EF_i + FS(i,j), \\ EF_j = ES_j + T_j. \end{cases} \tag{8.53}$$

3. 结束到结束关系，如图 8.19(b)，

$$\begin{cases} EF_j = EF_i + FF(i,j), \\ ES_j = EF_j - T_j. \end{cases} \tag{8.54}$$

4. 开始到结束关系，如图 8.19(d)，

$$\begin{cases} EF_j = ES_i + SF(i,j), \\ ES_j = EF_j - T_j. \end{cases} \tag{8.55}$$

5. 如果工序 j 的前导关系不止一个，则应分别计算然后取最大值，并称取得最大值的那个前导关系为**起作用的**.

第 2 步　整个工程的开始与结束.

1. 如果出现某个工序 i 的最早开始日期 $ES_i \leqslant 0$，意味着工序 i 在整个工程开始时即可开工. 此时可从开始工序 0 引一虚箭头指向工序 i，即把开始工序作为 i 的紧前工序.

2. 设 $EF_k = \max\{EF_i | 0 < i < n\}$，则工序 k 的完工意味着整个工程的结束，故 EF_k 就是总工期，此时可从工序 k 引一虚箭头指向结束工序 n，即把结束工序作为 k 的紧后工序. 由此，

$$T_E = \max\{EF_i | 0 < i < n\}. \tag{8.56}$$

第 3 步　前导关系的**独立时差**.

1. SS 关系，如图 8.19(c)，

$$IF(i,j) = ES_j - ES_i - SS(i,j), \tag{8.57}$$

2. FS 关系，如图 8.19(a)，

$$IF(i,j) = ES_j - EF_i - FS(i,j), \tag{8.58}$$

3. FF 关系，如图 8.19(b)，

$$IF(i,j) = EF_j - EF_i - FF(i,j), \tag{8.59}$$

4. SF 关系，如图 8.19(d)，

$$IF(i,j) = EF_j - ES_i - SF(i,j). \tag{8.60}$$

第 4 步　工序的时差.

1. **局部时差**

$$r_i = \min\{IF(i,j) | (i,j) \text{ 是边}\}. \tag{8.61}$$

在式(8.61)中不用"$j \in \Gamma^+(i)$"而用"(i,j) 是边"，是由于独立时差 $IF(i,j)$ 是对每条以 i 为尾以 j 为头的边计算的.

2. **总时差**

由式(8.49)，并注意到总时差不能超过总工期与工序最早结束时间的差，故有递推公式

$$\begin{cases} R_n = 0, \\ R_i = \min\{\min\{IF(i,j) + R_j | (i,j) \text{ 是边}\}, T_E - EF_i\}. \end{cases} \tag{8.62}$$

第 5 步　工序的**最迟开始时间**和**最迟结束时间**.

$$LS_i = ES_i + R_i. \tag{8.63}$$

$$LF_i = EF_i + R_i. \tag{8.64}$$

第 6 步　**关键路**.

从开始工序，沿总时差为零的工序及起作用的前导关系的边到结束工序，构成关键路.

例 8.25　已知工序的前导关系如表 8.7，试画出前导网络，并计算时间参数，确定关键路.

表 8.7　工序表

工序	工时	前导关系
A	10	
B	8	$SS(A,B)=3$
C	20	$FF(A,C)=2$
D	6	$SF(B,D)=5$
E	12	$FF(B,E)=5$，紧前工序 A
F	14	$SS(E,F)=2$，$FF(E,F)=5$，紧前工序 D
G	2	$FS(D,G)=4$，$SS(F,G)=3$，$SS(C,G)=10$

引入开始工序 S 与结束工序 T，画出前导网络如图 8.22.

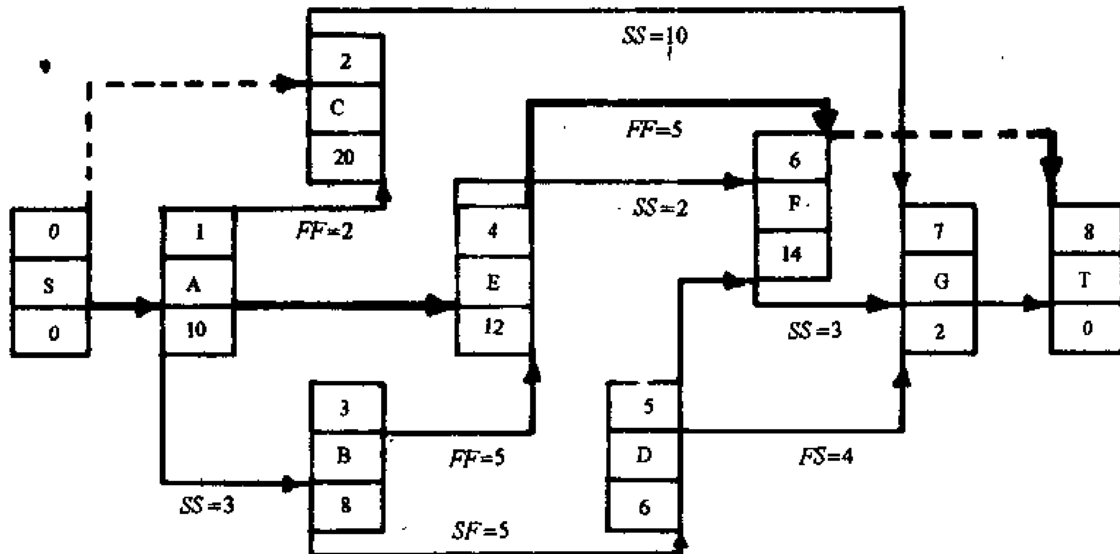


图 8.22　前导网络

(一) 计算 ES，EF 和 T_E

(1) $ES_1=0$，$EF_1=10$.

(2) $EF_2=EF_1+FF(1,2)=12$，$ES_2=EF_2-T_2=-8$.

∴　　工序 C 在整个工程开始时即可开工，即令

$$ES_2=0,\ EF_2=ES_2+T_2=20,$$

并以开始工序 S 为工序 C 的紧前工序，从顶点 0 向顶点 2 引一条虚边，如图 8.22.

(3) $ES_3=ES_1+SS(1,3)=3$，　$EF_3=ES_3+T_3=11$.

(4) $EF_4=EF_3+FF(3,4)=16$，　$ES_4=EF_4-T_4=4$，

　$ES_4=EF_1=10$，　　　　　$EF_4=ES_4+T_4=22$.

取大者得 $ES_4=10$，$EF_4=22$.

所有计算结果如表 8.8.

表 8.8　ES 与 EF

序号	1	2	3	4	5	6	7
ES	0	0	3	10	2	13	16
EF	10	20	11	22	8	27	18
起作用	0	1	1	1	3	4,FF	6

因为 $EF_6=27$ 是所有 EF 的最大值，所以

$$T_E=27.$$

并以结束工序 T 为工序 F 的紧后工序，从顶点 6 向顶点 8 引一条虚边，如图 8.22.

（二）独立时差，如表 8.9

表 8.9　独立时差

边	(0,1)	(0,2)	(1,2)	(1,3)	(1,4)	(2,7)	(3,4)	(3,5)
IF	0	0	8	0	0	6	6	0

边	(4,6),SS	(4,6),FF	(5,6)	(5,7)	(6,7)	(6,8)	(7,8)
IF	1	0	5	4	0	0	9

（三）时差与 LS,LF

利用表 8.9，由式(8.61) 可算得局部时差，再计算总时差，LS 与 LF，结果如表 8.10.

表 8.10

i	0	1	2	3	4	5	6	7	8
r_i	0	0	6	0	0	4	0	9	0
R_i	0	0	7	5	0	5	0	9	0
LS_i	0	0	7	8	10	7	13	25	27
LF_i	0	10	27	16	22	13	27	27	27

（四）关键路

关键路从开始工序 0，沿总时差为零的工序 1,4 与 6 到结束工序 8，由于 4 到 6 起作用的关系是 FF，故可在网络图中标出关键路，如图 8.22 中粗线所示.

§8.8　应用——非肯定型工程网络

有些复杂工程或大型工程的工序时间很难确切地给出，即工序时间具有不肯定性. 在这种情形，我们把工序时间看作是随机变量. 工序时间是随机变量的工程网络称为**非肯定型工程网络**.

对工序时间的估计一般采用三种时间：

(1) **乐观时间** a，即在最顺利的情形下工序的完成时间.

(2) **保守时间** b，即在最不利的情形下工序的完成时间.

(3) **最可能时间** m，即在通常情形下工序的完成时间.

假定工序时间这个随机变量服从以 a,b 为参数的 β 分布，则由概率论可知随机变量的

期望值 μ 和方差 σ^2 分别为

$$\mu=\frac{a+4m+b}{6}. \tag{8.65}$$

$$\sigma^2=\left(\frac{b-a}{6}\right)^2. \tag{8.66}$$

把工序 i 的期望值作为工序时间 T_i，方差为 σ_i，由此即可画出网络图并计算时间参数.

因为工序时间是随机变量，所以总工期也是随机变量. 假定工序时间是相互独立的，则当关键路 P 上的工序较多时，总工期 T_E 是服从正态分布 $N(E(T_E),D(T_E))$ 的随机变量，其中 $E(T_E)$ 是总工期 T_E 的期望值，$D(T_E)$ 是方差，并且

$$E(T_E)=\sum\{T_i \mid i\in P\}, \tag{8.67}$$

$$D(T_E)=\sum\{\sigma_i^2 \mid i\in P\}, \tag{8.68}$$

式中 $i\in P$ 表 i 是关键路 P 上的工序即关键工序.

$E(T_E)$ 是总工期的期望值. 假定希望总工期 T_E 不超过 T_c，则其概率为

$$\begin{aligned}P(T_E\leqslant T_c)&=\int_{-\infty}^{T_c}\frac{1}{\sqrt{2\pi D(T_E)}}\exp\left(-\frac{1}{2}\left(\frac{t-E(T_E)}{\sqrt{D(T_E)}}\right)^2\right)\mathrm{d}t\\&=\int_{-\infty}^{\frac{T_c-E(T_E)}{\sqrt{D(T_E)}}}\frac{1}{2\pi}\exp\left(-\frac{1}{2}y^2\right)\mathrm{d}y.\end{aligned}$$

令

$$X=\frac{T_c-E(T_E)}{\sqrt{D(T_E)}}, \tag{8.69}$$

$$\Phi(X)=\int_{-\infty}^{X}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}y^2\right)\mathrm{d}y, \tag{8.70}$$

则 X 服从标准正态分布 $N(0,1)$. 因此，当给定 T_c 之后，由式(8.69) 计算 X，再由标准正态分布表查 $\Phi(X)$，就知道整个工程在不超过 T_c 时完成的概率.

从正态分布表可查得 X 与 $\Phi(X)$ 的关系，如表 8.11.

表 8.11

X	−3	−0.5	0	0.5	3
$\Phi(X)$	0.00	0.30	0.50	0.69	0.99

表 8.11 指出，当 $X\leqslant-3$ 时，$\Phi(X)\approx 0$，即在预计完成时司 T_c 内完成整个工程几乎是不可能的. 当 $-3<X<-0.5$ 时，$0<\Phi(X)<0.3$，即在预计 T_c 时间内完成整个工程是困难的，这样的网络计划偏于冒进. 当 $-0.5\leqslant X\leqslant 0.5$ 时，$0.3\leqslant\Phi(X)\leqslant 0.69$，即在预计时间 T_c 内完成整个工程是可能的，这样的网络计划是较合适的. 当 $X>0.5$ 时，$\Phi(X)>0.69$，即在预计时间 T_c 内完成整个工程是容易的，这样的网络计划偏于保守.

算法 8.9　非肯定型工程网络的算法

第 1 步　确定工序，工序的乐观时间、保守时间、最可能时间，各工序的紧前工序.

第 2 步　计算工序时间的期望值和方差.

第 3 步　画出工序网络图，工序时间用期望值代替.

第 4 步　按算法 8.7 进行时间参数计算.

第 5 步　在第 4 步的总工期实际上是总工期的期望值 $E(T_E)$，再由式(8.68) 计算总工期的方差 $D(T_E)$.

第 6 步　以 T_c 表示整个工程的预计完成时间，把 $E(T_E)$ 与 $D(T_E)$ 代入式(8.69).

(1) 由 $X \leqslant -3$ 解得的 T_c，在预计时间 T_c 内完成整个工程几乎是不可能的.

(2) 由 $-3 < X < -0.5$ 解得的 T_c，在预计时间 T_c 内完成整个工程是困难的，网络计划偏于冒进.

(3) 由 $-0.5 \leqslant X \leqslant 0.5$ 解得的 T_c，在预计时间 T_c 内完成整个工程是很可能的，网络计划比较合适.

(4) 由 $X > 0.5$ 解得的 T_c，在预计时间 T_c 内完成整个工程是容易的，网络计划偏于保守.

第 7 步　当给定整个工程的预计完成时间 T_c 之后，可由式(8.69) 算出 X，然后由正态分布 $N(0,1)$ 表查出 $\Phi(X)$，即得总工期不超过 T_c 时完成整个工程的概率.

例 8.26　已知工序表 8.12，试画出工序网络图，计算时间参数，确定关键路，求出较合适网络计划的 T_c，又问当 $T_c = 44$ 时，整个工程完成的概率是多少？

表 8.12　工序表

工序	乐观时间	最可能时间	保守时间	紧前工序
A	2	5	8	/
B	4	7	10	A
C	6	9	12	A
D	8	8	8	A
E	3	12	21	B
F	1	4	7	B、C
G	5	14	17	D、F
H	3	6	9	E、G
I	5	8	11	H

首先由式(8.65) 与式(8.66) 计算期望值与方差，如表 8.13.

表 8.13　期望值与方差

工序	A	B	C	D	E	F	G	H	I
期望值	5	7	9	8	12	4	13	6	8
方差	1	1	1	0	9	1	4	1	1

其次用期望值作为工期画出工序网络图，然后进行时间参数计算，用粗线标出关键路，并把所有结果标在网络图中，如图 8.23.

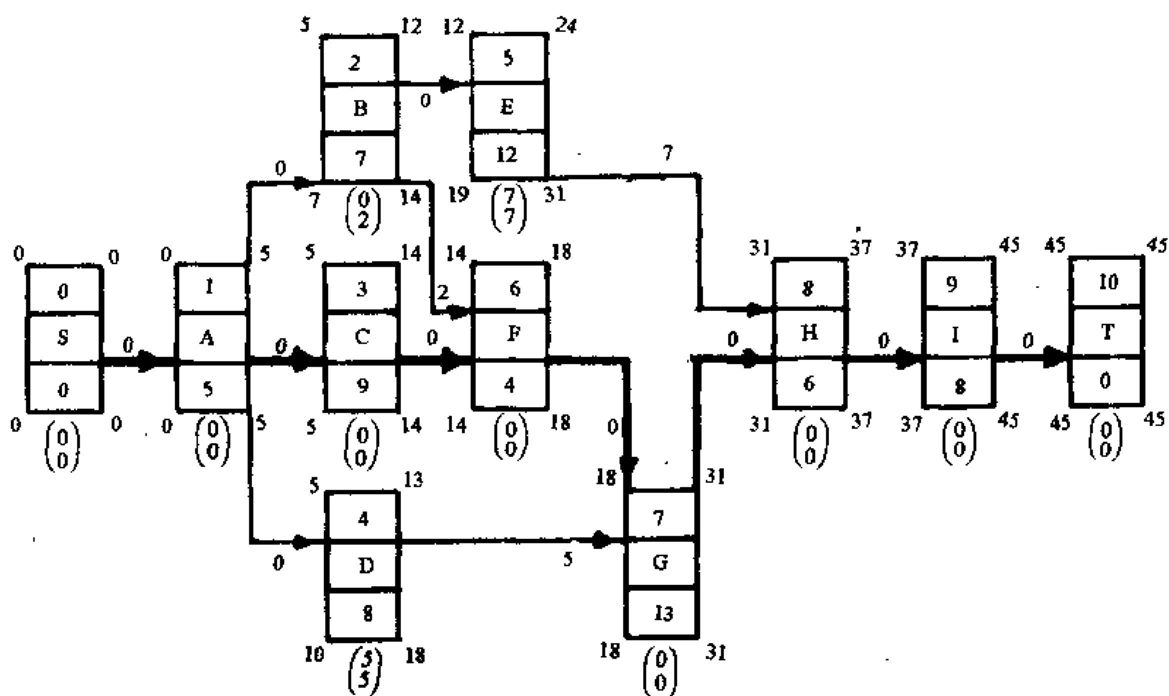


图 8.23

关键路是 SACFGHIT. 由式(8.67) 与式(8.68),

$$E(T_E)=45,\ D(T_E)=9,$$

代入式(8.69), $X=\dfrac{T_c-45}{3}$,并由 $-0.5\leqslant X\leqslant 0.5$ 解得

$$43.5\leqslant T_c\leqslant 46.5,$$

即当预计完成时间在 43.5 与 46.5 之间时,网络计划是比较合适的.

当 $T_c=44$ 时,代入式(8.69), $X=-\dfrac{1}{3}$,查正态分布表, $\Phi(-\dfrac{1}{3})\approx 0.37$,即整个工程在总工期不超过 44 时完成的概率是 37%.

习　题　八

8.1　设 f 是网络 N 的一个流, $S\subseteq V$,

(1) 举例说明 $f^+(S)\neq\sum\limits_{v\in S}f^+(v)$,

$$f^-(S)\neq\sum_{v\in S}f^-(v),$$

(2) 证明 $\sum\limits_{v\in S}(f^+(v)-f^-(v))=f^+(S)-f^-(S)$.

8.2　证明引理 8.1.

8.3　设网络 $N=(V,E,c,X,Y)$ 转化为等价网络 $N'=(V',E',c',s,t)$, f 是 N 的流,令

$$f'(e)=\begin{cases}f(e) & ,e\in E,\\ f^+(x)-f^-(x), & e=(s,x),\\ f^-(y)-f^+(y), & e=(y,t),\end{cases}$$

(1) 证明 f' 是 N' 的流，

(2) 证明 $\text{val}\, f' = f'^{+}(s) = f'^{-}(t) = \text{val}\, f$.

8.4　给定网络如图 8.24，试列出所有的流，并由此求出最大流.

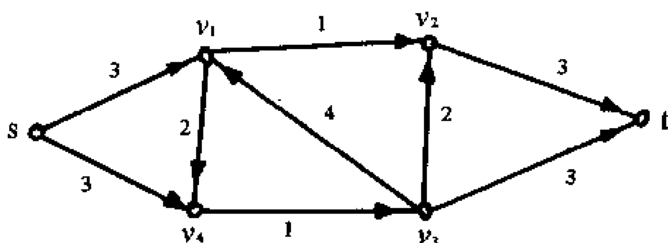


图 8.24

8.5　设有 q 台机器 $B_1,\cdots,B_q$，有 p 件加工任务 $A_1,\cdots,A_p$. 若已知每台机器能够完成那些加工任务的表，问能同时加工的任务最多有多少?试建立网络模型.

8.6　对图 8.2 的网络，利用定理 8.3 求最大流和最小割.

8.7　对习题 8.4 的网络，试列出所有的割，并由此求出最小割和最大流.

8.8　试建立求偶图最大匹配的网络模型.

8.9　在网络 $N=(V,E,c,s,t)$ 中若没有有向 s-t 路，则最大流是零流.

8.10　最大流问题有有限解的充要条件是，不存在有向 s-t 路而各边的容量都是 ∞.

8.11　证明引理 8.4.

8.12　若 $(A,\overline{A})$，$(B,\overline{B})$ 都是网络 N 的最小割，则

$$(A\cup B,\overline{A\cup B})\text{ 与 }(A\cap B,\overline{A\cap B})$$

也都是 N 的最小割.

8.13　用算法 8.1 求图 8.3 的网络的最大流.

8.14　证明引理 8.6.

8.15　证明引理 8.7.

8.16　证明引理 8.8.

8.17　证明引理 8.10.

8.18　用算法 8.2 求出如图 8.25 所给定网络的最大流与最小割，设初始流为零流.

8.19　给定正整数 λ，能求得网络 N 的流 f 使 $\text{val}\, f=\lambda$ 吗?试修改算法 8.2 求给定值 λ 的流.

8.20　对习题 8.18 的网络，试用习题 8.19 中提出的算法求流值为 6 的流，设初始流为零流.

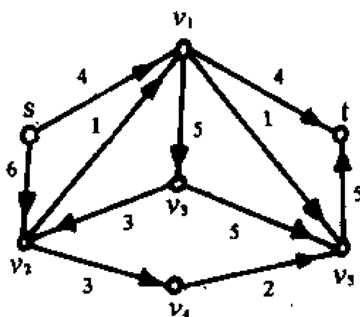


图 8.25

8.21　列出图 8.8(a) 的网络中所有流值为 4 的流，并验证图 8.8(d) 的流是流值 4 的最小代价流.

8.22　证明引理 8.11.

8.23　证明引理 8.12.

8.24　证明定理 8.13.

8.25　给定如图 8.26 的网络，利用算法 8.3 求流值 5 的最小代价流.

8.26　对习题 8.25 的网络，用算法 8.3 求最小代价最大流.

8.27　对习题 8.25 的网络，用算法 8.4 求流值 5 的最小代价流.

8.28　对习题 8.25 的网络，用算法 8.4 求最小代价最大流.

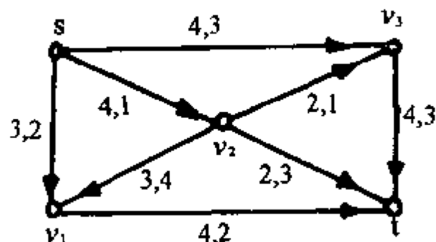


图 8.26

8.29　如图 8.27，河的两岸为敌对双方部队所占领，河中间有 4 个岛，有 14 座桥，问至少要截断几座桥 B 岸部队才能阻止 A 岸部队过桥追击，试用网络方法求解.

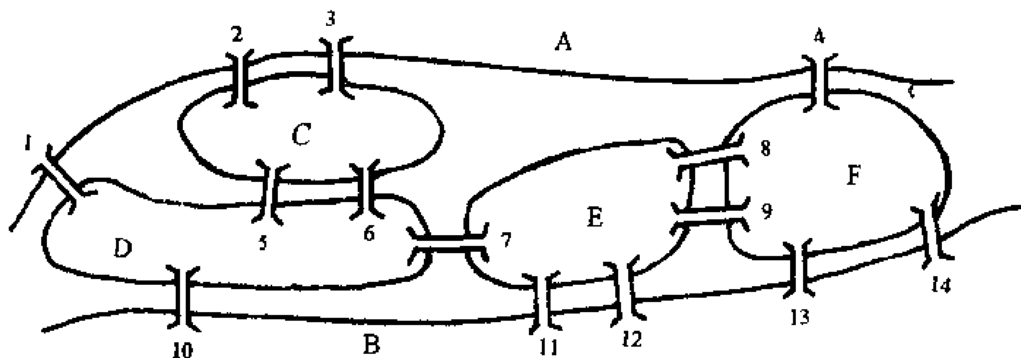


图 8.27

8.30　要从 3 个仓库运送某商品到 4 个零售点，已知供、需量及各仓库到零售点的容量如下表. 问现有运送渠道能否满足零售点的需要，试用网络方法求解.

容量 零售点 / 仓库	B_1	B_2	B_3	B_4	供应量
A_1	30	10	0	40	20
A_2	0	0	10	50	20
A_3	20	10	40	5	100
需求量	20	20	60	20	

8.31　下面是一个多阶段存贮模型. 某公司根据市场情况预计某商品在未来 6 个月的需要量，进货单价与存贮单价如下表. 当月订购当月所需商品不需要存贮费用. 试用网络的方法确定进货与存贮的合理水平使总费用最少，并列出逐月进货，存贮与费用表.

月　份	1	2	3	4	5	6
需要量(件)	50	55	50	45	40	30
进货单价(元)	825	775	850	850	775	825
存贮单价(元)	40	30	35	20	40	0

8.32　现有4个工人加工4种零件其收益如下表，试用网络的方法安排每个工人恰好加工一种零件并使收益最低者的收益最大.

收益＼零件 工人	B_1	B_2	B_3	B_4
A_1	5	9	2	7
A_2	3	4	8	5
A_3	6	7	7	3
A_4	6	10	3	4

8.33　某厂需要某种设备，若购置新设备则需购置费，若继续使用旧设备则需维修费.预计5年内购置与维修费如下表，试用网络方法制订一个购置与维修的5年计划使总费用最少.

购置年	1	2	3	4	5
购置费(万元)	20	21	23	24	26
使用年限	1	2	3	4	5
维修费(万元)	8	13	19	23	30

8.34　某厂有2个车间——部件车间与组装车间，先由部件车间生产部件，再由组装车间组装成成品.部件车间每月最多生产10个部件，其产品送人库房，每月初组装车间到库房领取部件，预计今后5个月对部件的需求量与生产每个部件所耗工时如下表.假定开始和第5月末的库存数都是零，试用网络方法求解下列问题.

(1) 列出工时最少的部件生产计划，

(2) 若库房容量为9个部件，试列出工时最少的部件生产计划.

月	1	2	3	4	5
需求量(件)	—	8	5	7	4
工时(每件小时)	11	16	13	17	—

8.35　用网络方法求解习题4.23的问题.

8.36　证明定理8.15.

8.37　用例8.14的Z与F验证式(8.29).

8.38　给定开关网络如图8.28，写出连接矩阵，传输矩阵并验证式(8.29).

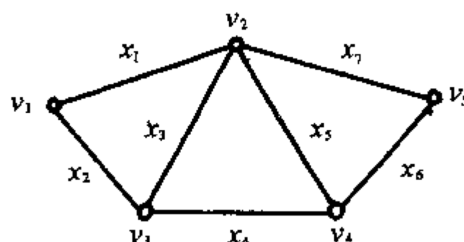


图8.28

8.39 (1) 对不可分二端图,如何从路矩阵求圈矩阵,给出算法,

(2) 如习题 8.38 的网络,写出 v_1 与 v_5 的路矩阵,并由(1)的算法求圈矩阵.

8.40 举例说明,对不是不可分二端图,定理 8.16 不成立.

8.41 如习题 8.38 的开关网络,由 v_1 与 v_5 的路矩阵求基本路矩阵并写出基本路.

8.42 给定开关函数

$$f_{ij} = x_1x_2x_3x_5x_7 + x_1x_3x_4x_6 + x_1x_5x_6x_8 + x_2x_4 + x_2x_3x_5x_8 + x_3x_4x_6x_7x_8 + x_5x_6x_7,$$

求单接触网络实现.

8.43 改正下面如图 8.29 的工程网络.

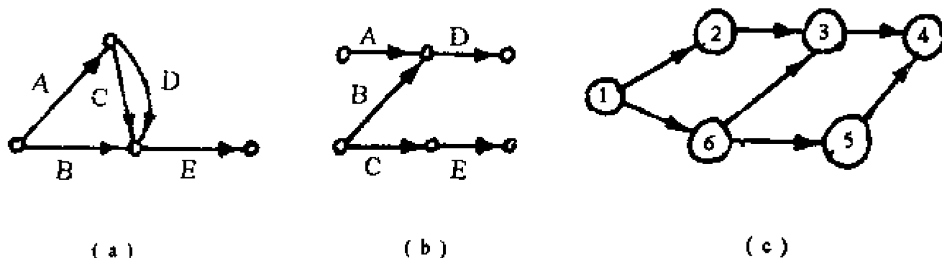


图 8.29

8.44 给出工序及紧前工序表,试画出工程网络图,并给事项编号.

工序	A	B	C	D	E	F	G	H
紧前工序	/	A	A,B	A,B	C	C	B	D,E,G

8.45 证明引理 8.18.

8.46 证明引理 8.19.

8.47 给出工序表,试画出事项网络,并计算事项与工序的时间参数,然后找出关键路,把这些结果都画在网络图上.

工序	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
工时	12	3	5	7	8	5	6	4	3	8	3	7	3	2	2
紧前工序	/	/	B	B	B	C	C	A G	D E	E	H	F	I	J	K L M

8.48 为缩短总工期,对表 8.2 作如下变更:

(1) 工序 C 分解为两个平行工序 C_1 与 C_2,其工时分别是 4 与 3,

(2) 工序 E 分解为衔接的 3 个子工序 E_1,E_2 与 E_3,其工时分别为 4,2 与 4,而工序 E_1 与 E 的紧前工序 B 交叉作业,工序 E_3 与 E 的紧后工序 G 交叉作业.

试进行网络时间参数计算,然后画出网络图并与图 8.18 的网络图加以比较.

8.49 把习题 8.43 中改正后的事项网络改画为工序网络.

8.50 由习题 8.44 的工序表画出工序网络.

8.51 给出工序表如习题 8.47,试画出工序网络,并计算时间参数,找出关键路,然后把这些结果都标在网络图上,再与事项网络相比较.

8.52 给出工序表如下,试画出工序网络,并计算时间参数,找出关键路,再把这些

结果都标在网络图上.

工序	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
工时	4	2	2	5	12	8	4	4	12	12	12	20	8	8	16	10
紧前工序	/	/	/	/	C	A	B	E	L M	M N	N O	P	H	G	F	D

8.53　对习题 8.52 的工序表进行事项网络计算并把结果标在网络图中.

8.54　已知下列工序表,试画出前导网络,计算时间参数,确定关键路,并把结果标在网络图上.

工序	工时	前导关系
A	3	/
B	7	紧前工序 A,F
C	20	紧前工序 B
D	24	$SS(C,D)=10$, $FF(C,D)=10$, $SS(I,D)=1$, $SS(J,D)=5$
E	20	紧前工序 D
F	2	/
G	5	紧前工序 B,F
H	25	紧前工序 G
I	50	$SS(C,I)=10$, $FF(C,I)=10$
J	50	$SS(I,J)=5$, $SS(H,J)=5$, $FF(H,J)=5$
K	16	紧前工序 J
L	2	紧前工序 E

8.55　已知工序的三种估计时间及紧前工序如下表.

工序表

工序	乐观时间	最可能时间	保守时间	紧前工序
A	2	3	4	/
B	1	4	7	A
C	2	2	8	A
D	16	18	20	B
E	16	22	22	B
F	4	4	10	B,C
G	15	15	15	B,C
H	18	18	18	D,E,F
I	5	6	13	G
J	4	4	4	H
K	9	10	11	I

(1) 计算工序时间的期望值与方差，

(2) 画出工序网络图，

(3) 计算时间参数，

(4) 确定关键路，

(5) 计算总工期的期望值与方差，

(6) 求比总工期期望值提前 2 个单位时间完成的概率，

(7) 求比总工期期望值延期 1 个单位时间完成的概率，

(8) 求比较合适网络计划的 T_c.

第九章　网络规划

§9.1　网络规划

图论中的网络问题，诸如运输问题，转运问题，分配问题，最大流问题，最短路问题，最小费用流问题等都能转化为线性规划问题，称为**网络规划**. 也可以说，网络规划是线性规划在网络中的应用. 由于网络规划的特性，解线性规划的单纯形法在求解网络规划时有着更为简单而有效的形式.

例 9.1（转运问题）　假定有 m 个仓库 $A_1,\cdots,A_m$，有 n 个商店 $B_1,\cdots,B_n$，有 r 个转运站 $C_1,\cdots,C_r$. 已知仓库 A_i 的供应量 a_i，商店 B_j 的需求量 b_j，并且总供应量等于总需求量，即

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{9.1}$$

仓库，商店与转运站之间有路可通，每段路的运输单价已知，问如何把 m 个仓库的货物全部运送到 n 个商店，使总运费最少？

今以 X 记 m 个仓库的顶点，称为发点. 以 Y 记 n 个商店的顶点，称为收点. 以 I 记 r 个转运站的顶点，称为中间点. 则得顶点集

$$V = X \cup Y \cup I.$$

若从 i 到 j 有路，则对应边 $e=(i,j)$，边的权就是运输单价. 由此得到一个赋权有向图 $G=(V,E)$ 满足

(1) $X,Y \subseteq V$, $X \cap Y = \varnothing$, $X \neq \varnothing$, $Y \neq \varnothing$ 且

$\forall\ i \in X$ 有一个供应量 $a_i \geqslant 0$ 整数，

$\forall\ j \in Y$ 有一个需求量 $b_j \geqslant 0$ 整数，

合于条件(9.1).

(2) 边的容量不受限制.

(3) $\forall\ e \in E$ 对应运输的单位成本 $c(e) \geqslant 0$.

问题是求每条边 e 的运输量 $f(e)$ 使总运费最少，即求解下列线性规划，称为**转运网络规划**.

$$\min \sum_{e\in E} c(e)f(e) \tag{9.2}$$

$$\begin{aligned}
\text{s.t.}\ & \sum_{e\in X^+(i)} f(e) - \sum_{e\in X^-(i)} f(e) = a_i,\ i \in X, \\
& \sum_{e\in X^+(i)} f(e) - \sum_{e\in X^-(i)} f(e) = -b_i,\ i \in Y, \\
& \sum_{e\in X^+(i)} f(e) - \sum_{e\in X^-(i)} f(e) = 0,\ i \in I,
\end{aligned}$$

$$f(e) \geqslant 0,\ e \in E.$$

设 $e = (i,j)$，可令 $x_{ij} \triangleq f(e)$，$c_{ij} \triangleq c(e)$，则得 $|E|$ 维列向量 $x = (x_{ij})_{(i,j)\in E}$ 与 $c = (c_{ij})_{(i,j)\in E}$. 再令

$$b_i = \begin{cases} i\text{的供应量} & ,i \in X, \\ -(i\text{的需求量}) & ,i \in Y, \\ 0 & ,i \in I, \end{cases}$$

则得 $|V|$ 维列向量 $b = (b_i)_{i\in V}$. 于是转运网络规划(9.2)可写为矩阵形式

$$\begin{aligned} &\min\ cx \\ &\text{s.t.}\ \ Ax = b, \\ &\qquad\ x \geqslant 0, \end{aligned} \tag{9.3}$$

其中 A 是 $G = (V,E)$ 的关联矩阵，而平衡条件(9.1)成为

$$\sum_{i\in V} b_i = 0.$$

例 9.2(运输问题) 设某产品有 m 个产地，产地 i 的产量是 a_i，有 n 个销地，销地 j 的销量是 b_j，且产销平衡，即

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

已知从产地 i 到销地 j 的运输单价 c_{ij}，如何确定运输方案，把产品从产地运到销地，使总运费最少？

以 X 表 m 个产地的顶点，Y 表 n 个销地的顶点，边 (i,j) 的权是运输单价 c_{ij}，我们得到无向图 $G = (X,Y,E)$ 是一个赋权完全偶图. 问题是求每条边 (i,j) 的运输量 x_{ij} 使总运费最少，即求解下列线性规划，称为**运输网络规划**.

$$\begin{aligned} &\min\ \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &\text{s.t.}\ \ \sum_{j=1}^{n} x_{ij} = a_i,\ i = 1,\cdots,m, \\ &\qquad\ \sum_{i=1}^{m} x_{ij} = b_j,\ j = 1,\cdots,n, \\ &\qquad\quad x_{ij} \geqslant 0 \quad \forall\ i,j. \end{aligned} \tag{9.4}$$

令

$$\begin{aligned} x &= (x_{11},\cdots,x_{1n},\cdots,x_{m1},\cdots,x_{mn})^T, \\ c &= (c_{11},\cdots,c_{1n},\cdots,c_{m1},\cdots,c_{mn}), \\ b &= (a_1,\cdots,a_m,b_1,\cdots,b_n)^T, \end{aligned}$$

A 是关联矩阵，

则运输网络规划(9.4)的矩阵形式仍然是(9.3).

例 9.3(分配问题) 假定分配 n 个人作 n 件工作，人 i 作工作 j 的效益是 c_{ij}，问如何分配工作使总效益最大？

以 X 表人的顶点集，以 Y 表工作的顶点集，边 (i,j) 的权就是人 i 作工作 j 的效益 c_{ij}，则得无向图 $G = (X,Y,E)$ 是一个赋权完全偶图.

令

$$x_{ij}=\begin{cases}1, & \text{人 } i \text{ 作工作 } j,\\ 0, & \text{否则},\end{cases} \tag{9.5}$$

分配问题就是求解下列线性规划，称为**分配网络规划**.

$$\begin{aligned}&\max \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &s.t.\ \sum_{j=1}^{n} x_{ij}=1,\ i=1,\cdots,n, \\ &\qquad \sum_{i=1}^{n} x_{ij}=1,\ j=1,\cdots,n, \\ &\qquad x_{ij}=0 \text{ 或 } 1\quad \forall\ i,j.\end{aligned} \tag{9.6}$$

令

$$\begin{aligned}&x=(x_{11},\cdots,x_{1n},\cdots,x_{n1},\cdots,x_{nn})^T,\\ &c=(c_{11},\cdots,c_{1n},\cdots,c_{n1},\cdots,c_{nn}),\\ &b=(\underbrace{1,\cdots,1}_{n\text{个}},\underbrace{1,\cdots,1}_{n\text{个}})^T,\\ &\qquad A \text{ 是关联矩阵},\end{aligned}$$

则分配网络规划(9.6)的矩阵形式是

$$\begin{aligned}&\max\ cx \\ &s.t.\ Ax=b, \\ &\qquad x \text{ 是 0-1 向量}.\end{aligned} \tag{9.7}$$

容易看到，分配网络规划是运输网络规划的特例，运输网络规划是转运网络规划的特例.

例 9.4(**最大流问题**)　设 $N=(V,E,c,s,t)$ 是一个网络，最大流问题是求解下列问题：

$$\max\ \{\mathrm{val}\ f \mid f \text{ 是 } N \text{ 的流}\}.$$

设 A 是图 $G=(V,E)$ 的关联矩阵，令

$$b_i=\begin{cases}\lambda &, i=s,\\ -\lambda &, i=t,\\ 0 &, i\in V\backslash\{s,t\}.\end{cases}$$

再设 $e=(i,j)$，可令 $x_{ij}\triangleq f(e)$，得 $|E|$ 维列向量 $x=(x_{ij})_{(i,j)\in E}$，则最大流问题就是求解下列线性规划，称为**最大流网络规划**.

$$\begin{aligned}&\max\ \lambda \\ &s.t.\ Ax=b, \\ &\qquad 0\leqslant x\leqslant c.\end{aligned}$$

在网络图 $G=(V,E)$ 中附加一条边 (t,s)，称为**返回边**，返回边的容量为 ∞. 记新的网络为

$$\overline{N}=(V,\overline{E},\tilde{c},s,t),$$

其关联矩阵为 $\overline{A}$，由引理 8.4 和定理 3.7，最大流网络规划的等价形式是

$$\max x_{lu} \tag{9.8}$$
$$s.t.\ \bar{A}\bar{x} = 0,$$
$$0 \leqslant \bar{x} \leqslant \bar{c},$$

其中 $\bar{x} = \begin{pmatrix} x \\ x_{lu} \end{pmatrix}$.

例 9.5(最短路问题) 设 $G = (V,E)$ 是有向图,每条边 e 有一个权 $c(e) \geqslant 0$,求顶点 v_0 到顶点 v_t 的最短路问题是求解下列问题:

$$\min \{c(P) \mid P \text{ 是 } v_0\text{-}v_t \text{ 路}\},$$

其中

$$c(P) = \sum_{e \in E(P)} c(e).$$

设图 G 的关联矩阵是 A,令

$$b_i = \begin{cases} 1 & ,i = v_0, \\ -1 & ,i = v_t, \\ 0 & ,i \in V \setminus \{v_0, v_t\}, \end{cases}$$

$$x_j = \begin{cases} 0, & \text{最短路不过边 } j, \\ 1, & \text{否则}, \end{cases}$$

则最短路问题就是求解下列线性规划,称为**最短路网络规划**.

$$\min cx \tag{9.9}$$
$$s.t.\ Ax = b,$$
$$x \text{ 是 0-1 向量}.$$

例 9.6(最小费用流问题) 设 $G = (V,E)$ 是一个有向图,每个顶点 $i \in V$ 对应一个数 b_i. 若 $b_i > 0$,称 i 为**发点**;若 $b_i < 0$,称 i 为**收点**;若 $b_i = 0$,称 i 为**中间点**,且满足平衡条件 $\sum_{i \in V} b_i = 0$. 每条边 $e \in E$ 对应二个数,一个是容量 $u(e)$,另一个是运输单价 $c(e)$. 最小费用流就是从发点把货物全部运送到收点使总运输费最小. 换言之,即求解下列线性规划,称为**最小费用流网络规划**.

$$\min cx \tag{9.10}$$
$$s.t.\ Ax = b,$$
$$0 \leqslant x \leqslant u,$$

其中 A 是图 G 的关联矩阵.

很明显,转运网络规划,运输网络规划,分配网络规划,最大流网络规划与最短路网络规划都是最小费用流网络规划的特例. 最小费用流网络规划实际上是转运网络规划加上上界条件. 对单纯形法稍加修改即可解上界问题(参见[13]),因此,我们以转运网络规划作为网络规划的一般形式来讨论.

§9.2 解的整数性

在 §9.1 中看到,网络规划的系数矩阵 A 就是有向图或偶图的关联矩阵. 因此,由习题 3.21 与习题 3.23,我们得到网络规划的系数矩阵是全单模的.

引理 9.1　设 A 是 $m\times n$ 矩阵，I 是 $m\times m$ 单位矩阵，则 A 是全单模的充要条件是，(AI) 是全单模的(习题 9.7).

引理 9.2　设 $S=\{x\mid Ax=b,x\geqslant 0\}$，其中 $A\in R^{m\times n}$，$b\in R^m$，都是整数的，且 $r(A)=m$. 若 A 是全单模的，则 S 的基可行解都是整数的(习题 9.8).

定理 9.3　设 $A\in R^{m\times n}$ 是整数的，$r(A)=m$，则下列 4 条等价.

(1) A 是全单模的；

(2) 若 $b\in R^m$ 是整数的，则

$$S=\{x\mid Ax\leqslant b,\ x\geqslant 0\}$$

的极点是整数的；

(3) (AI) 的每个基 B 的逆是整数的；

(4) A 的每个非奇异子矩阵的逆是整数的.

证　(1)⇒(2)　由引理 9.1 与引理 9.2.

(2)⇒(3)　设 B 是基，取 $a\in R^m$ 是整数的，使

$$a+(B^{-1})_{\cdot i}\geqslant 0.$$

令 $x_B=a+(B^{-1})_{\cdot i}$，$b=Ba+I_{\cdot i}$，则

(Ⅰ) $b\in R^m$ 是整数的，

(Ⅱ) $Bx_B=b$，

(Ⅲ) $x_B\geqslant 0$，

即 x_B 是 $S=\{x\mid Ax\leqslant b,x\geqslant 0\}$ 的极点，由条件(2)，x_B 是整数的. 故

$$(B^{-1})_{\cdot i}=x_B-a$$

是整数的. 从而推出 B^{-1} 是整数的.

(3)⇒(4)　设 F 是 A 的任一非奇异子矩阵，令

$$B=\begin{pmatrix}F & 0\\ M & I\end{pmatrix},$$

其中 I 是单位阵，则 B 是 (AI) 的基，由条件(3)，

$$B^{-1}=\begin{pmatrix}F^{-1} & 0\\ -MF^{-1} & I\end{pmatrix}$$

是整数的，故 F^{-1} 是整数的.

(4)⇒(1)　设 F 是 A 的任一非奇异子矩阵，由条件(4)，F^{-1} 是整数的，故 $\det F$，$\det F^{-1}$ 都是整数，又

$$\det FF^{-1}=\det F\det F^{-1}=1,$$

所以 $|\det F|=1$. 因此 A 是全单模的.

定理证毕.

定理 9.4　(1) 分配网络规划，最短路网络规划的基可行解是整数的.

(2) 若容量是整数的，则最大流网络规划的基可行解是整数的.

(3) 若供应量与需求量是整数的，则转运网络规划与运输网络规划的基可行解是整数的.

(4) 若供应量，需求量与容量是整数的，则最小费用流网络规划的基可行解是整数

的.

证 因为关联矩阵 A 是全单模的,故由引理 9.2 与定理 9.3 得证. **证毕.**

在下面的讨论中,我们总假定供应量,需求量与容量都是整数的,因而网络规划的基可行解也都是整数的. 解的整数性是网络规划的一个重要性质. 另一个重要性质是,解网络规划的单纯形法只需加减运算而不需乘除运算. 这一性质读者将在以后各节看到.

§9.3 运输网络规划

运输网络规划的一般形式是式(9.4),它是产销平衡的问题. 如果产销不平衡,则可化为平衡问题.

(1) 若产大于销,即 $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j > 0$,此时可设虚销地 $(n+1)$,其销量 b_{n+1} 为

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j,$$

并令 $c_{i,n+1} = 0, i = 1,\cdots,m$,意为从产地 i 运到虚销地 $(n+1)$ 的量就是留在产地 i 未运出的量,故运输单价为零.

(2) 若产小于销,即 $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j < 0$,此时可设虚产地 $(m+1)$,其产量 a_{m+1} 为

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i,$$

并令 $c_{m+1,j} = 0, j = 1,\cdots,n$,意为从虚产地 $(m+1)$ 运到销地 j 的量就是未运到销地 j 的量,即销地 j 供应不足的量,故运输单价为零.

因此,我们只讨论平衡的运输问题.

为方便,把运输网络的系数矩阵记为

$$A = (P_{11},\cdots,P_{1n},\cdots,P_{m1},\cdots,P_{mn}),$$

$$P_{ij} = I_{\cdot i} + I_{\cdot j} \quad (I \text{ 是 } m+n \text{ 维单位阵}).$$

引理 9.5 运输网络规划是可行的.

证 容易看到,

$$x_{ij} = \frac{a_i b_j}{\sum_{i=1}^{m} a_i}, \ i = 1,\cdots,m, \ j = 1,\cdots,n$$

是一个可行解. **证毕.**

引理 9.6 运输网络规划有最优解.

证 因为

$$0 \leqslant x_{ij} \leqslant \min\{a_i, b_j\},$$

所以运输网络规划有界. 由线性规划基本定理,可行且有界,则必有最优解. **证毕.**

下面用单纯形法来解运输网络规划. 首先用运输表来描述运输网络是方便的: 在表 9.1 中,行表示产量平衡,列表示销量平衡.

运输表中方格 (i,j) 称为点. 点 (i,j) 对应网络图的边 (i,j),也对应关联矩阵 A 的列 P_{ij}.

表 9.1　运输表

产地 \ 销地	1	……	j	……	n	产量
1	c_{11}　x_{11}		c_{1j}　x_{1j}		c_{1n}　x_{1n}	a_1
⋮						⋮
i	c_{i1}　x_{i1}		c_{ij}　x_{ij}		c_{in}　x_{in}	a_i
⋮						⋮
m	c_{m1}　x_{m1}		c_{mj}　x_{mj}		c_{mn}　x_{mn}	a_m
销量	b_1	……	b_j	……	b_n	

运输网络图是一个偶图，由定理 1.10，偶图不含奇圈，故运输网络的圈必为偶圈. 容易看到，在网络图中的圈

$$i_1 j_1 i_2 j_2 \cdots i_{k-1} j_{k-1} i_k j_k i_1 \ (k \geqslant 2) \tag{9.11}$$

对应运输表中闭回路

$$(i_1, j_1),\ (i_2, j_1),\ (i_2, j_2),\ \cdots,\ (i_{k-1}, j_{k-1}),$$
$$(i_k, j_{k-1}),\ (i_k, j_k),\ (i_1, j_k),\ (i_1, j_1),\quad (k \geqslant 2). \tag{9.12}$$

例 9.7　表 9.2 的运输表中，闭回路

$$(1,2),(2,2),(2,5),(3,5),(3,3),(1,3),(1,2)$$

对应网络图 9.1 中的圈

$$x_1 y_2 x_2 y_5 x_3 y_3 x_1.$$

表 9.2　闭回路

	1	2	3	4	5	6
1		x_{12}	x_{13}			
2		x_{22}			x_{25}	
3			x_{33}		x_{35}	
4						

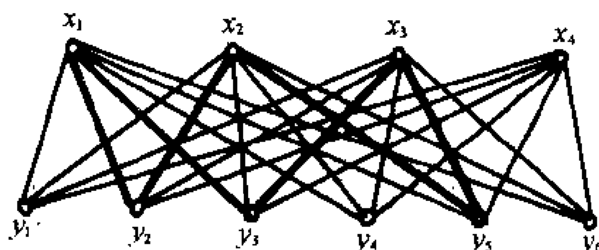


图 9.1　闭回路对应的圈（图中用粗线表示）

引理 9.7　运输网络规划的 $(m+n)$ 个平衡条件不是独立的，但任意 $(m+n-1)$ 个平衡条件都是独立的（习题 9.11）.

设想增加一个虚销地 $(n+1)$，销量为零，并增加一条边 $(1, n+1)$，这条边对应的变量 $x_{1,n+1}$ 称为**人造变量**，由此得到的网络称为**人造运输网络**，其关联矩阵是 $\hat{A}$，

$$\hat{A} = \begin{pmatrix} A & I_{\cdot 1} \\ 0 & 1 \end{pmatrix}, \tag{9.13}$$

其中 $I_{\cdot 1}$ 是 $(m+n)$ 阶单位阵 I 的第 1 列. 人造运输网络有 $(m+n+1)$ 个平衡条件，由引理 9.7，其中任意 $(m+n)$ 个平衡条件是独立的. 去掉关于虚销地 $(n+1)$ 的平衡条件，得人

造运输网络规划的等价问题

$$\min cx \tag{9.14}$$

$$s.t.\ (A\quad I_{\cdot 1})\begin{pmatrix} x \\ x_{1,n+1} \end{pmatrix} = b,$$

$$\begin{pmatrix} x \\ x_{1,n+1} \end{pmatrix} \geqslant 0.$$

引理 9.8 B 是人造运输网络规划(9.14)的基,当且仅当 B 包含 A 的 $(m+n-1)$ 个无关列加上人造变量 $x_{1,n+1}$ 对应的列 $I_{\cdot 1}$(习题 9.12).

引理 9.8 指出,$I_{\cdot 1}$ 始终是人造运输网络规划的基向量,故人造变量 $x_{1,n+1}$ 始终是取零值的基变量. 因此,略去基向量 $I_{\cdot 1}$,我们说运输网络规划的基由 A 的 $(m+n-1)$ 个无关列组成. 这 $(m+n-1)$ 个无关列对应的基变量正好是运输表中的 $(m+n-1)$ 个点. 事实上,由于虚销地 $(n+1)$ 的平衡条件被删去,因此在运输表中并不出现人造变量的点.

现在的问题是,对运输表中的 $(m+n-1)$ 个点,如何判断所对应的 A 的列是否构成基.

引理 9.9 设 M 是运输表的一个非空点集,则

$$\{P_{ij} \mid (i,j) \in M\}$$

线性无关的充要条件是,M 不含闭回路(习题 9.13).

定理 9.10 $\{P_{ij} \mid (i,j) \in M\}$ 是基的充要条件是,M 含 $(m+n-1)$ 个点且不含闭回路.

证 运输网络的基由 A 的 $(m+n-1)$ 个无关列组成,再由引理 9.9 得证.

定义 9.1 设 M 是运输表的一个点集,点 $(i,j) \in M$ 称为**孤立点**,若 M 中没有与 (i,j) 同行或同列的点.

引理 9.11 设 M 是运输表的一个非空点集,

(1) 若 M 是闭回路,则 M 没有孤立点;

(2) 若 M 没有孤立点,则 M 含闭回路;

(习题 9.14).

引理 9.12 设 $\{P_{ij} \mid (i,j) \in M\}$ 是基,P_{kl} 是非基向量,则 $M \cup \{(k,l)\}$ 有唯一闭回路,且点 (k,l) 在该闭回路上(习题 9.15).

设 $B = \{P_{ij} \mid (i,j) \in M\}$ 是基,则非基向量 P_{kl} 可由基 B 唯一表出. 由引理 9.12,若能求出 $M \cup \{(k,l)\}$ 的唯一闭回路,则可相应得出 P_{kl} 由基 B 的唯一表示式. 这样的闭回路可利用引理 9.11,反复去掉孤立点即可求得.

算法 9.1 求运输表的闭回路的算法

设 $B = \{P_{ij} \mid (i,j) \in M\}$ 是基,P_{kl} 是非基变量.

令 $M = M \cup \{(k,l)\}$,$M_1 = \varnothing$.

第 1 步 任取 $(i,j) \in M$.

第 2 步 若 $\{(i,t) \in M \mid t \neq j\} \neq \varnothing$,转第 4 步.

第 3 步 去掉孤立点 (i,j),即令

$$M = M\backslash\{(i,j)\},$$

转第 1 步.

第 4 步　若$\{(t,j)\mid t \neq i\} = \varnothing$,转第 3 步.

第 5 步　令 $M_1 = M_1 \cup \{(i,j)\}$. 若 $M\backslash M_1 = \varnothing$,停,已得含$(k,l)$的唯一闭回路 M_1. 否则,任取$(i,j) \in M\backslash M_1$,转第 2 步.

运输网络规划的初始基可行解

以上讨论我们得到:$\{x_{ij}\mid(i,j) \in M\}$ 是基可行解的充要条件是,

(1) $|M| = m + n - 1$,并且

(2) M 不含闭回路,并且

(3) 满足运输表的 m 个行平衡即产量平衡与 n 个列平衡即销量平衡.

西北角法正好可利用上面三个条件求出初始基可行解. 所谓**西北角法**,就是每次给运输表的西北角元(即左上角元) 一个值使其满足行平衡(产量平衡) 或列平衡(销量平衡),然后划去已满足平衡条件的行或列. 于是每确定一个西北角元的值就划去一行或一列. 当确定了$(m + n - 2)$个西北角元之后,运输表还剩下一个点未划去. 由于产销平衡,最后一个点的值唯一确定,并同时划去一行与一列. 由此,总共确定了$(m + n - 1)$ 个点的值. 因为每次所产生的点没有同行或同列的点,所以这$(m + n - 1)$ 个点不含闭回路. 故这$(m + n - 1)$ 个点构成初始基可行解.

算法 9.2　西北角法——求运输问题的初始基可行解

以 $I = \{1,\cdots,m\}$ 表产地的标号,以 $J = \{1,\cdots,n\}$ 表销地的标号,以 S 表基变量的集.

令 $x_{ij} = 0, i \in I, j \in J, S = \varnothing$.

第 1 步　确定西北角元,令

$$r = \min\ \{i \mid i \in I\},$$
$$t = \min\ \{j \mid j \in J\}.$$

第 2 步　确定西北角元的值,令

$$k = \min\ \{a_r, b_t\}, x_{rt} = k,$$
$$a_r = a_r - k\quad, b_t = b_t - k,$$
$$S = S \cup \{x_{rt}\}.$$

第 3 步　删去已经满足的平衡条件,

$$若\ a_r = 0,\ 令\ I = I\backslash\{r\},$$
$$若\ b_t = 0,\ 令\ J = J\backslash\{t\}.$$

第 4 步　若 $a_r + b_t = 0$ 即同时删去行标 r 与列标 t,转第 5 步;否则转第 1 步.

第 5 步　若 $I = \varnothing$,停,S 为基可行解的变量集. 否则,令

$$S = S \cup \{x_{r,t+1}\},$$

转第 1 步.

注　在第 5 步,若 $I \neq \varnothing$,为保证"每确定一个西北角元就划去一行或一列"的原则,应补充一个取零值的基变量,它可以是 $x_{r,t+1}$,也可以是 $x_{r+1,t}$,前者正是在第 5 步中所作的. 对后者,可在第 5 步中令

$$S = S \cup \{x_{r+1,t}\}.$$

例 9.8 给定运输表如表 9.3，用西北角法求初始基可行解.

表 9.3 运输表

4	7	11	20	20
10	5	8	7	14
6	2	12	10	9
4	15	16	5	12
10	9	15	21	

解 (1) $r = 1,\ t = 1,\ k = 10,\ x_{11} = 10,$

$a_1 = 10,\ b_1 = 0,\ S = \{x_{11}\}$，删去第 1 列.

(2) $r = 1,\ t = 2,\ k = 9,\ x_{12} = 9,$

$a_1 = 1,\ b_2 = 0,\ S = \{x_{11}, x_{12}\}$，删去第 2 列.

(3) $r = 1,\ t = 3,\ k = 1,\ x_{13} = 1,$

$a_1 = 0,\ b_3 = 14,\ S = \{x_{11}, x_{12}, x_{13}\}$，删去第 1 行.

(4) $r = 2,\ t = 3,\ k = 14,\ x_{23} = 14,$

$a_2 = 0,\ b_3 = 0,$

$S = \{x_{11}, x_{12}, x_{13}, x_{23}\}$，删去第 2 行与第 3 列，

$S = \{x_{11}, x_{12}, x_{13}, x_{23}, x_{24}\}.$

(5) $r = 3,\ t = 4,\ k = 9,\ x_{34} = 9,$

$a_3 = 0,\ b_4 = 12,$

$S = \{x_{11}, x_{12}, x_{13}, x_{23}, x_{24}, x_{34}\}$，删去第 3 行.

(6) $r = 4,\ t = 4,\ k = 12,\ x_{44} = 12,$

$a_4 = 0,\ b_4 = 0,$

$S = \{x_{11}, x_{12}, x_{13}, x_{23}, x_{24}, x_{34}, x_{44}\}$，删去第 4 行与第 4 列. 因为 $I = \varnothing$，停. 已得初始基可行解：

$$x_{11} = 10,\ x_{12} = 9,\ x_{13} = 1,\ x_{23} = 14,\ x_{24} = 0,\ x_{34} = 9,\ x_{44} = 12.$$

计算过程与结果可列表如下：

表 9.4 西北角法

4 10	7 9	11 1	20	20,10,1,0
10	5	8 14	7 0	14,0
6	2	12	10 9	9,0
4	15	16	5 12	12,0
10 0	9 0	15 14 0	21 12 0	

西北角法的优点是直观，简单；缺点是没有考虑运输单价的影响. 把西北角法中的原

则“每次给西北角元一个值”改为“每次给单位运价最小的元一个值”，由此即得求初始基可行解的另一方法——**最小元素法**. 一般说，最小元素法求得的解比西北角法求得的解其目标值更小. 如本例，由西北角法所求的解，目标值为 376. 若用最小元素法则所得解的目标值是 331，比用西北角法所求得的值少 45(习题 9.18).

运输网络规划的判别数

人造运输网络规划(9.14)有$(m+n)$个约束条件，其单纯形乘子即对偶变量可设为

$$w=(u_1,\cdots,u_m,v_1,\cdots,v_n),$$

因为人造变量 $x_{1,n+1}$ 始终是取零值的基变量，故运输单价 $c_{1,n+1}$ 可为任意常数. 设

$$B=\{P_{ij}\mid(i,j)\in M\}\cup\{I_{\cdot 1}\},$$

$$c_B=\{c_{ij}\mid(i,j)\in M\}\cup\{c_{1,n+1}\},$$

则 $wB=c_B$ 的分量形式是

$$\begin{cases}u_i+v_j=c_{ij},(i,j)\in M,\\ u_1=c_{1,n+1},\end{cases}$$

不妨设 $c_{1,n+1}=0$ 则有

$$\begin{cases}u_i+v_j=c_{ij}, & (i,j)\in M,\\ u_1=0,\end{cases}\tag{9.15}$$

我们称单纯形乘子 w 的$(m+n)$个分量 u_i 与 v_j 为$(m+n)$个顶点的**位势**. 利用位势可求出修正单纯形法的判别数 $r=wN-c_N$，其分量形式为

$$r_{ij}=u_i+u_j-c_{ij},\ (i,j)\notin M.\tag{9.16}$$

例 9.9 对表 9.3 的运输规划，在表 9.4 中求出了基可行解，今由式(9.15)求位势如表 9.5.

表 9.5 位势

c_{ij} \ v_j / u_i	4	7	11	10
0	4	7	11	
−3			8	7
0				10
−5				5

再由式(9.16)求非基变量判别数(基变量的判别数为零，故只需求非基变量的判别数)如表 9.6.

表 9.6 判别数

r_{ij} \ v_j / u_i	4	7	11	10
0				−10
−3	−9	−1		
0	−2	5	−1	
−5	−5	−13	−10	

关于新的基可行解

当由式(9.16)求出判别数后，若 $r \leqslant 0$，则当前解为最优解. 否则令

$$r_{kl} = \max\ \{r_{ij} \mid (i,j) \notin M\} > 0, \tag{9.17}$$

此时 P_{kl} 进入基. 由引理 9.12，$M \cup \{(k,l)\}$ 有唯一闭回路. 利用算法 9.1 得此闭回路，设为 C，并把 C 的顶点分为两类：C^+ 与 C^-.

(1) $(k,l) \in C^+$，

(2) 在 C 中与 (k,l) 同行的元 $\in C^-$，

(3) 在 C 中与(2)的元同列的元 $\in C^+$，

(4) 继续上述过程，直至把 C 分解为 C^+ 与 C^-. 显然，C 中同一行或同一列的两个顶点恰好一个在 C^+，另一个在 C^-.

因为 $\sum\{P_{ij} \mid (i,j) \in C^+\} = \sum\{P_{ij} \mid (i,j) \in C^-\}$，所以非基向量 P_{kl} 由基 B 的唯一表示式是

$$P_{kl} = \sum_{(i,j)\in C^-} P_{ij} - \sum_{(i,j)\in C^+\setminus\{(k,l)\}} P_{ij}. \tag{9.18}$$

由单纯形表的性质，P_{kl} 由基 B 表出的系数即 $B^{-1}P_{kl}$，故由式(9.18)得

$$(B^{-1}P_{kl})_{(i,j)} = \begin{cases} 1 & ,(i,j) \in C^-, \\ -1, & (i,j) \in C^+ \setminus\{(k,l)\}, \\ 0 & ,\text{其它}. \end{cases} \tag{9.19}$$

于是，由确定出基向量的最小比值法则，取 θ，

$$\theta = x_{rt} = \min\ \{x_{ij} \mid (i,j) \in C^-\}, \tag{9.20}$$

则 P_{rt} 退出基，且新的基可行解是 $\{x'_{ij} \mid (i,j) \in M'\}$，

$$x'_{ij} = \begin{cases} x_{ij} + \theta, (i,j) \in C^+, \\ x_{ij} - \theta, (i,j) \in C^-, \\ x_{ij} \qquad ,\text{其它}. \end{cases} \tag{9.21}$$

$$M' = (M \cup \{(k,l)\})\setminus\{(r,t)\}. \tag{9.22}$$

以上讨论得到解运输网络规划的单纯形法. 由于 $(m+n)$ 个对偶变量称为位势，所以这样的方法称为**位势法**.

算法 9.3　位势法——解运输网络规划的单纯形法

给定运输表如表 9.1.

第 1 步　求初始基可行解 $\{x_{ij} \mid (i,j) \in M\}$，可用西北角法或最小元素法.

第 2 步　求位势 u_i 与 v_j，由式(9.15).

第 3 步　求判别数 r_{ij}，由式(9.16).

第 4 步　最优性判别.

　　若 $r_{ij} \leqslant 0$，则当前解为最优解.

第 5 步　找入基向量 P_{kl}，由式(9.17).

第 6 步　求闭回路 C，由算法 9.1.

第 7 步　求出基向量 P_{rt}.

分解 C 为 C^+ 与 C^-，再由式(9.20)求 θ 与 P_n.

第 8 步　求新的基可行解，由式(9.21)与式(9.22). 令 $x_{ij}=x'_{ij}$，$M=M'$，转第 2 步.

例 9.10　求解表 9.3 的运输网络规划.

从算法 9.3，解运输网络规划的单纯形法——位势法可在表上进行. 表 9.7 是第 1 次迭代，数据是利用表 9.4～表 9.6 的结果. 在表 9.7 中，方框左上角为运输单价 c_{ij}，右下角为非基变量判别数 r_{ij}，中间为基变量 x_{ij} 的值. x_{kl} 为进入基的变量. x_{ij} 右上角的"＋"或"－"表示属于闭回路 C 的 C^+ 或 C^-. 表 9.8 是第 2 次迭代.

表 9.7　第 1 次迭代

		1	2	3	4	
	u_i \ v_j	4	7	11	10	a_i
1	0	[4] 10	[7] 9^-	[11] 1^+	[20] -10	20
2	-3	[10] -9	[5] -1	[8] 14^-	[7] 0^+	14
3	0	[6] -2	[2] x_{32}^+ 5	[12] -1	[10] 9^-	9
4	-5	[10] -5	[15] -13	[16] -10	[5] 12	12
	b_j	10	9	15	21	

表 9.8　第 2 次迭代

		1	2	3	4	
	u_i \ v_j	4	7	11	10	a_i
1	0	[4] 10	[7] 0	[11] 10	[20] -10	20
2	-3	[10] -9	[5] -1	[8] 5	[7] 9	14
3	-5	[6] -7	[2] 9	[12] -6	[10] -5	9
4	-5	[10'] -11	[15] -13	[16] -10	[5] 12	12
	b_j	10	9	15	21	

因为 $r_{ij}\leqslant 0$，所以已得最优解，其基变量的值是

$$x_{11}=10,\ x_{12}=0,\ x_{13}=10,\ x_{23}=5,$$

$$x_{24}=9,\ x_{32}=9,\ x_{44}=12,$$

最优值是 331.

§9.4　分配网络规划

由网络规划解的整数性，分配网络规划(9.6)等价于

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} \tag{9.23}$$

$$\begin{aligned} s.t.\ & \sum_{j=1}^{n} x_{ij} = 1,\ i = 1,\cdots,n, \\ & \sum_{i=1}^{n} x_{ij} = 1,\ j = 1,\cdots,n, \\ & x_{ij} \geqslant 0 \quad \forall\ i,j. \end{aligned}$$

实际上，分配网络规划(9.23)就是当 $m = n, a_i = b_j = 1$ 的运输网络规划. 因此，解运输网络规划的位势法可用来解分配网络规划. 下面利用分配网络规划的特性讨论更为简单的求解方法.

不失一般性，我们讨论极小化问题

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} \tag{9.24}$$

$$\begin{aligned} s.t.\ & \sum_{j=1}^{n} x_{ij} = 1,\ i = 1,\cdots,n, \\ & \sum_{i=1}^{n} x_{ij} = 1,\ j = 1,\cdots,n, \\ & x_{ij} \geqslant 0 \quad \forall\ i,j. \end{aligned}$$

首先由线性规划的对偶理论，分配网络规划(9.24)的对偶规划是

$$\max \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j \tag{9.25}$$

$$\begin{aligned} s.t.\ & u_i + v_j \leqslant c_{ij}, \\ & i,j = 1,2,\cdots,n. \end{aligned}$$

设 x_{ij} 是原始可行解，u_i 与 v_j 是对偶可行解，由互补松驰定理，x_{ij}，u_i 与 v_j 分别是原始最优解与对偶最优解的充要条件是

$$\begin{aligned} & (c_{ij} - u_i - v_j)x_{ij} = 0, \\ & i,j = 1,2,\cdots,n. \end{aligned} \tag{9.26}$$

其次，我们用分配矩阵来描述分配网络规划是方便的. 分配矩阵就是分配网络规划(9.24)的目标函数的系数构成的矩阵

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}. \tag{9.27}$$

分配网络规划(9.24)的解，就是在分配矩阵(9.27)中选择 n 个元满足

(1) 这 n 个元位于分配矩阵的不同行不同列(可行性)，并且

(2) 这 n 个元的和最小(最优性).

引理 9.13 在分配矩阵(9.27)中任一行加或减同一常数 k，或者任一列加或减同一常数 k，分配网络规划(9.24)的最优解不变，而最优值相应增加或减少同一常数 k(习题 9.39).

引理 9.13 指出，在分配矩阵中对同一行或同一列加或减同一常数，其最优解是一致的，由此引入下面的定义.

定义 9.2 在分配矩阵(9.27)中，任一行加或减同一常数，或者任一列加或减同一常数，所得矩阵称为**等价分配矩阵**.

令

$$\hat{u}_i = \min\{c_{ij} \mid 1 \leqslant j \leqslant n\},\ i = 1,\cdots,n, \tag{9.28}$$

$$\hat{v}_j = \min\{c_{ij} - \hat{u}_i \mid 1 \leqslant i \leqslant n\},\ j = 1,\cdots,n, \tag{9.29}$$

$$\hat{c}_{ij} = c_{ij} - \hat{u}_i - \hat{v}_j,\ i,j = 1,\cdots,n. \tag{9.30}$$

则可得等价分配矩阵

$$\begin{pmatrix} \hat{c}_{11} & \hat{c}_{12} & \cdots & \hat{c}_{1n} \\ \hat{c}_{21} & \hat{c}_{22} & \cdots & \hat{c}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{c}_{n1} & \hat{c}_{n2} & \cdots & \hat{c}_{nn} \end{pmatrix}. \tag{9.31}$$

容易看到，矩阵(9.31)每行每列都至少有一个零元.

定义 9.3 在等价分配矩阵中位于不同行且不同列的零元称为**独立零元**. 划去矩阵中零元所在行的横线或划去零元所在列的竖线称为**覆盖线**. 有覆盖线的行(列)称为**覆盖行(列)**. 没有覆盖线的行(列)称为**未覆盖行(列)**. 矩阵中的元若只有一条覆盖线通过，则称为**覆盖元**. 若有两条覆盖线通过，则称为**重覆盖元**. 若没有覆盖线通过，则称为**未覆盖元**

引理 9.14 设 x_{ij} 是分配网络规划(9.24)的可行解，若

$$x_{ij} = 1 \Longrightarrow \hat{c}_{ij} = 0, \tag{9.32}$$

则 x_{ij} 是最优解.

证 首先，由式(9.28)与(9.29)得到的 $\hat{u}_i, \hat{v}_j$ 是对偶规划(9.25)的可行解. 其次，由式(9.32)，互补松驰条件(9.26)成立. 由线性规划的对偶定理，x_{ij} 是最优解. **证毕.**

若等价分配矩阵有 n 个独立零元，则独立零元相应的 x_{ij} 取值 1，其余的 x_{ij} 取值零，正好是分配网络规划(9.24)的可行解，且满足式(9.32)，故由引理 9.14，x_{ij} 是最优解. 换言之，求解分配网络规划就是在等价分配矩阵中找出 n 个独立零元.

例 9.11 已知分配矩阵

$$\begin{pmatrix} 3 & 2 & 5 & 4 \\ 1 & 2 & 3 & 4 \\ 6 & 3 & 1 & 5 \\ 2 & 5 & 3 & 4 \end{pmatrix}, \tag{9.33}$$

求最优分配.

在矩阵(9.33)中，先减去行最小，再减去列最小，得等价分配矩阵

$$\begin{pmatrix} 1 & \boxed{0} & 3 & 0 \\ \boxed{0} & 1 & 2 & 1 \\ 5 & 2 & \boxed{0} & 2 \\ 0 & 3 & 1 & \boxed{0} \end{pmatrix},$$

其中有方框的是独立零元，故得最优分配：

$$1 \to 2,\ 2 \to 1,\ 3 \to 3,\ 4 \to 4,$$

相应最小费用是 $2 + 1 + 1 + 4 = 8$.

问题是当独立零元个数小于 n 时应如何处理，我们有下面的定理.

定理 9.15 在等价分配矩阵中，独立零元的最大个数等于覆盖线的最少条数.

证 设 A 是分配网络图的关联矩阵，P_{ij} 是 A 的列向量. 今考虑下面一对对偶问题 (P) 与 (D)，其中

$$M = \{(i,j) \mid \hat{c}_{ij} = 0\},$$

e 为分量全为 1 的 $2n$ 维列向量.

$$(P) \qquad \begin{aligned} &\max \sum_{(i,j)\in M} x_{ij} \\ &s.t. \sum_{(i,j)\in M} x_{ij} P_{ij} \leqslant e, \\ &\qquad x_{ij} \geqslant 0,\ (i,j) \in M. \end{aligned}$$

$$(D) \qquad \begin{aligned} &\min \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j \\ &s.t.\ u_i + v_j \geqslant 1,\ (i,j) \in M, \\ &\qquad u_i, v_j \geqslant 0,\ i,j = 1,\cdots,n. \end{aligned}$$

显然，$x_{ij} = 0, (i,j) \in M$ 是 (P) 的可行解. $u_i = v_j = 1, i,j = 1,\cdots,n$ 是 (D) 的可行解，由对偶定理，(P) 与 (D) 都有最优解且最优值相等. 又由定理 9.4，基可行解都是整数的. 因此，(P) 的最优解就是使最多的 x_{ij} 取值 1 的解，其最优值就是独立零元的最大数.

另一方面，(D) 的最优解就是使最少的 u_i, v_j 取值 1 的解，注意到当 $u_i = 1$ 时，对每个 $(i,j) \in M$ 的 j 都有 $v_j = 0$，故相当于划去第 i 行零元的覆盖线；当 $v_j = 1$ 时，对每个 $(i,j) \in M$ 的 i 都有 $u_i = 0$，故相当于划去第 j 列零元的覆盖线. 换言之，(D) 的最优值就是划去零元的覆盖线的最小数.

综上得证. **证毕.**

设

$k =$ 独立零元的最大数，

$P =$ 未覆盖行标集，$|P| = p$，

$Q =$ 未覆盖列标集，$|Q| = q$，

$R =$ 覆盖行标集，$\quad |R| = n - p$，

$S =$ 覆盖列标集，$\quad |S| = n - q$.

则由定理 9.15，有

$$k = (n - p) + (n - q). \tag{9.34}$$

引理 9.16 独立零元全由覆盖元组成(习题 9.40).

引理 9.17 设独立零元的最大数 $k < n$，令

$$c_0 = \min\{\hat{c}_{ij} \mid i \in P, j \in Q\}, \tag{9.35}$$

$$\bar{u}_i = \begin{cases} \hat{u}_i + c_0, & i \in P, \\ \hat{u}_i, & i \in R, \end{cases} \tag{9.36}$$

$$\bar{v}_j = \begin{cases} \hat{v}_j & , j \in Q, \\ \hat{v}_j - c_0, & j \in S, \end{cases} \tag{9.37}$$

$$\bar{c}_{ij} = c_{ij} - \bar{u}_i - \bar{v}_j. \tag{9.38}$$

证明(习题 9.41)：

(1) $\bar{u}_i, \bar{v}_j$ 是对偶可行解.

(2) 等价分配矩阵

$$\begin{pmatrix} \bar{c}_{11} & \bar{c}_{12} & \cdots & \bar{c}_{1n} \\ \bar{c}_{21} & \bar{c}_{22} & \cdots & \bar{c}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \bar{c}_{n1} & \bar{c}_{n2} & \cdots & \bar{c}_{nn} \end{pmatrix}, \tag{9.39}$$

保持每行每列都有零元.

(3) 等价分配矩阵(9.39)比(9.31)独立零元至少增加一个.

归纳上面的讨论我们有：

(Ⅰ) 若$(\hat{c}_{ij})$有 n 个独立零元，则已得最优分配，

(Ⅱ) 若$(\hat{c}_{ij})$的独立零元少于 n 个，则构造等价分配矩阵$(\bar{c}_{ij})$，其独立零元至少增加一个，

(Ⅲ) 重复步骤(Ⅱ)，有限步后必终止在步骤(Ⅰ). 这就是利用互补松驰性解分配网络规划的方法，一般称为匈牙利方法.

算法 9.4　匈牙利方法——解分配网络规划的互补松驰算法

给定分配矩阵(9.27).

第 1 步　求等价分配矩阵. 由式(9.28)，(9.29)与(9.30)得等价分配矩阵(9.31).

第 2 步　求等价分配矩阵(9.31)的独立零元，注意到独立零元每行每列只有一个，故可如下进行.

(1) 若某行只有一个零元，该零元画框，所在列的其它零元都划掉；

(2) 若某列只有一个零元，该零元画框，所在行的其它零元都划掉；

(3) 若每个零元都有同行同列的零元，则任取一个零元画框，所在行与列的其它零元都划掉；

(4) 所有画框零元为所求独立零元，设其指标集为 M.

第 3 步　最优性判别：

若 $|M| = n$，则

$$x_{ij} = 1,\ (i,j) \in M$$

为最优分配.

第 4 步　覆盖等价分配矩阵(9.31)的零元，注意到独立零元由覆盖元组成，故可如下进行.

(1) 封锁没有画框零元的行；

(2) 封锁封锁行中的未画框零元所在列；

(3) 封锁封锁列中的画框零元所在行；

(4) 重复(2) 与(3) 直至不能再封锁；

(5) 未封锁行与封锁列各有一条覆盖线.

第 5 步　调整等价分配矩阵.

由式(9.35),(9.36),(9.37) 与(9.38) 得调整后的等价分配矩阵(9.39).

令$(\hat{c}_{ij}) = (\bar{c}_{ij})$,转第 2 步.

引理 9.18　在第 5 步的调整,等价于对$(\hat{c}_{ij})$ 实施下列三个规则之一：

规则 1　未覆盖行的元减 c_0,覆盖列的元加 c_0.

规则 2　未覆盖列的元减 c_0,覆盖行的元加 c_0.

规则 3　未覆盖元减 c_0,重覆盖元加 c_0.

(习题 9.42).

例 9.12　给定下列分配矩阵,求最优分配.

$$\begin{bmatrix} 2 & 7 & 5 \\ 5 & 2 & 1 \\ 2 & 6 & 5 \end{bmatrix}.$$

1. 等价分配矩阵 $\begin{bmatrix} 0 & 4 & 3 \\ 4 & 0 & 0 \\ 0 & 3 & 3 \end{bmatrix}$,

2. 画框求独立零元 $\begin{bmatrix} \boxed{0} & 4 & 3 \\ 4 & \boxed{0} & \varnothing \\ \varnothing & 3 & 3 \end{bmatrix}$,

4. 封锁与覆盖(用"√" 表封锁,用竖线覆盖)

$$\begin{bmatrix} \boxed{0} & 4 & 3 \\ 4 & \boxed{0} & 0 \\ 0 & 3 & 3 \end{bmatrix}\begin{matrix} \checkmark \\ \\ \checkmark \end{matrix}, \quad \begin{bmatrix} 0 & 4 & 3 \\ 4 & 0 & 0 \\ 0 & 3 & 3 \end{bmatrix}.$$
$$\checkmark$$

5. 调整等价分配矩阵 $\begin{bmatrix} 0 & 1 & 0 \\ 7 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$,

2. 画框求独立零元 $\begin{bmatrix} \boxed{0} & 1 & \varnothing \\ 7 & \boxed{0} & \varnothing \\ \varnothing & \varnothing & \boxed{0} \end{bmatrix}$,

3. 已求得 3 个独立零元,故得最优分配：

$$1 \to 1,\ 2 \to 2,\ 3 \to 3.$$

引理 9.19　(1) 证明等价分配矩阵非负.

(2) 当$(\hat{c}_{ij})$ 的独立零元少于 n 时,调整后得等价分配矩阵$(\bar{c}_{ij})$,证明

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\bar{c}_{ij} < \sum_{i=1}^{n}\sum_{j=1}^{n}\hat{c}_{ij}. \tag{9.40}$$

(3) 若分配矩阵是整数，证明有限步必求得整数最优解.

(习题 9.43).

§ 9.5 转运网络规划

现在讨论比运输网络规划与分配网络规划更一般的问题——转运网络规划. 首先有下面的引理.

引理 9.20 转运网络规划(9.3)的$(m+n+r)$个平衡条件不是独立的，但任意$(m+n+r-1)$个平衡条件都是独立的(习题 9.47).

为方便，我们设发点，收点与中间点共有 n 个，并设转运网络图 $G=(V,E)$ 是(n,m)赋权有向图，其顶点的顺序按发点，中间点，收点排列. 并不妨仍以 A 表示去掉关联矩阵最后一行的基本关联矩阵，仍以 b 表右端去掉最后一个分量所得向量，则转运网络规划可写为

$$\begin{aligned} &\min\ cx \\ &s.t.\ Ax=b, \\ &\qquad x\geqslant 0. \end{aligned} \tag{9.41}$$

引理 9.21 设 B 是转运网络规划(9.41)的基，则存在置换矩阵 P 与 Q 使 PBQ 为上三角阵且主对角线元全为 ± 1(习题 9.48).

下面用修正单纯形法解(9.41)，并设 B 是可行基，$A=(B,N)$.

关于判别数

设 w 是单纯形乘子，由 $wB=c_B$ 右乘 Q 得

$$wBQ=c_BQ. \tag{9.42}$$

令 $w=uP$，则式(9.42) 成为

$$u(PBQ)=c_BQ, \tag{9.43}$$

因为 PBQ 是上三角阵，所以由式(9.43) 容易解出 u，从而得出判别数 $r=wN-c_N$.

关于迭代方向

设 k 是入基向量指标，则修正单纯形法的迭代方向是 d，

$$d\triangleq\begin{pmatrix} d_B \\ d_N \end{pmatrix}=\begin{pmatrix} -B^{-1}N_{\cdot k} \\ I_{\cdot k} \end{pmatrix}. \tag{9.44}$$

对 $Bd_B=-N_{\cdot k}$ 左乘 P 得

$$PBd_B=-PN_{\cdot k}. \tag{9.45}$$

令 $d_B=Qy$，则式(9.45) 成为

$$(PBQ)y=-PN_{\cdot k}. \tag{9.46}$$

因为 PBQ 是上三角阵，容易从式(9.46) 求出 y，从而得出迭代方向 d.

关于迭代步长

修正单纯形法的迭代步长是

$$\theta=\min\left\{\frac{x_{B_i}}{-d_{B_i}}\,\middle|\,d_{B_i}<0\right\}. \tag{9.47}$$

引理 9.22 迭代方向(9.44)的分量是 0 或 1 或 -1(习题 9.49).

由引理 9.22，式(9.47) 等价于

$$\theta = \min \{x_{B_i} | d_{B_i} = -1\}. \tag{9.48}$$

引理 9.23 若转运网络规划可行则必有最优解(习题 9.50).

以上分析得求解转运网络规划的算法.

算法 9.5 解转运网络规划的修正单纯形法

设 B 是(9.41)的可行基,x_B 为基可行解,$A=(B,N)$.

第 1 步 求置换矩阵 P 与 Q,使 PBQ 为上三角阵且主对角线元全为 ± 1.

第 2 步 最优性检验.求解

$$u(PBQ) = c_B Q, \tag{9.49}$$

再计算

$$w = uP, \tag{9.50}$$

$$r = wN - c_N, \tag{9.51}$$

若 $r \leqslant 0$,则当前解为最优解.

第 3 步 求入基向量.令

$$r_k = \max \{r_j | j \in N\}, \tag{9.52}$$

得入基向量为 $N_{\cdot k}$.

第 4 步 求出基向量.求解

$$(PBQ)y = -PN_{\cdot k}, \tag{9.53}$$

再计算

$$d_B = Qy, \tag{9.54}$$

$$\theta = x_{B_l} = \min \{x_{B_i} | d_{B_i} = -1\}, \tag{9.55}$$

得出基向量为 $B_{\cdot l}$.

第 5 步 $N_{\cdot k}$ 与 $B_{\cdot l}$ 交换,从 B 得 B',从 N 得 N'.相应于 B' 的基可行解是

$$x' = \begin{pmatrix} x_B \\ 0 \end{pmatrix} + \theta \begin{pmatrix} d_B \\ I_{\cdot k} \end{pmatrix},$$

即

$$x_{B_i}' = \begin{cases} x_{B_i} + \theta d_{B_i}, & i \neq l, \\ 0, & i = l. \end{cases} \tag{9.56}$$

令 $B = B'$,$N = N'$,$x_B = x_B'$ 返回第 1 步.

例 9.13 某公司有 2 个工厂,2 个转运站,3 个零售商店,如图 9.2.顶点旁的数字为正的是工厂产量,为负的是商店需求量,为零的是转运仓库(数零不必在图中标出).箭杆旁的数字表示单位运价.求最优调运方案.

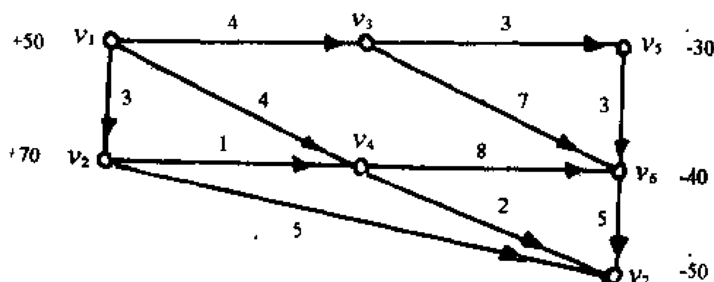


图 9.2

去掉最后一个顶点——收点 v_7 的平衡条件，得转运网络规划：

$$
\begin{aligned}
\min\ & 3x_{12}+4x_{13}+4x_{14}+x_{24}+5x_{27}+3x_{35} \\
& +7x_{36}+8x_{46}+2x_{47}+3x_{56}+5x_{67} \qquad (9.57)\\
\text{s.t.}\ & x_{12}+x_{13}+x_{14} = 50 \\
& -x_{12}+x_{24}+x_{27} = 70 \\
& -x_{13}+x_{35}+x_{36} = 0 \\
& -x_{14}-x_{24}+x_{46}+x_{47} = 0 \\
& -x_{35}+x_{56} = -30 \\
& -x_{36}-x_{46}-x_{56}+x_{67} = -40, \\
& x_{ij} \geqslant 0.
\end{aligned}
$$

其系数矩阵与右端是

$$
\begin{array}{c|ccccccccccc}
 & (1,2) & (1,3) & (1,4) & (2,4) & (2,7) & (3,5) & (3,6) & (4,6) & (4,7) & (5,6) & (6,7) \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 1
\end{array} = A,
$$

$$
b = \begin{pmatrix} 50 \\ 70 \\ 0 \\ 0 \\ -30 \\ -40 \end{pmatrix}.
$$

由定理 3.16 和定理 3.19，B 是基的充要条件是，B 对应于转运网络图的生成树. 我们不妨就以生成树的边集表示基 B. 换言之，在网络图中，B 表生成树；在网络规划中，B 表基，即基矩阵或基向量的集. 这样的表示既方便又不致发生混淆. 在图 9.3 中画出了转运网络图 9.2 的生成树，由此得系数矩阵 A 的分解 $A=(B,N)$，

$$B=\{(1,3),(2,4),(2,7),(3,5),(3,6),(4,6)\},$$

$$N=\{(1,2),(1,4),(4,7),(5,6),(6,7)\},$$

相应基可行解 $x_B=(50,20,50,30,20,20)^T$.

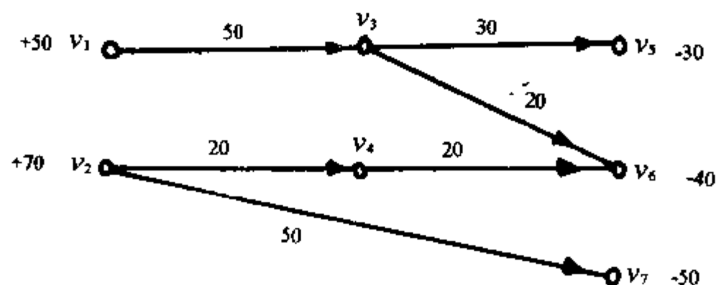


图 9.3

第 1 次迭代：

1. 求得

$$P=\begin{pmatrix}0&1&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&1\\0&0&1&0&0&0\\0&0&0&0&1&0\\1&0&0&0&0&0\end{pmatrix}\quad 与 \quad Q=\begin{pmatrix}0&0&0&0&0&1\\0&1&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&0\\0&0&1&0&0&0\end{pmatrix},$$

使

$$PBQ=\begin{pmatrix}1&1&0&0&0&0\\0&-1&1&0&0&0\\0&0&-1&-1&0&0\\0&0&0&1&1&-1\\0&0&0&0&-1&0\\0&0&0&0&0&1\end{pmatrix}.$$

2. $r=(-1,-1,2,1,-9)$.

3. $k=3, N_{\cdot 3}$ 进入基.

4. $d_B=(0,1,-1,0,0,0)^T$.

$\theta=x_{B_3}=50,\ l=3,\ B_{\cdot 3}$ 退出基.

5. $B=\{(1,3),(2,4),(4,7),(3,5),(3,6),(4,6)\}$,

$N=\{(1,2),(1,4),(2,7),(5,6),(6,7)\}$,

$x_B=(50,70,50,30,20,20)^T$. 如图 9.4.

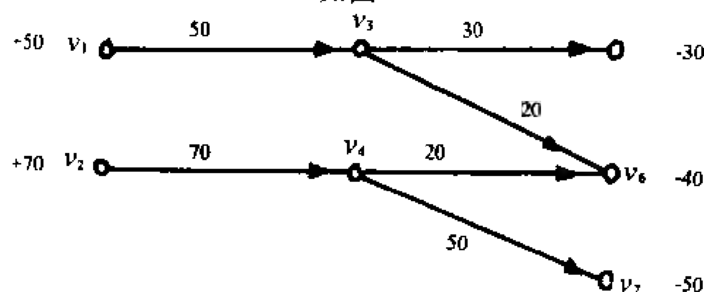


图 9.4

第 2 次迭代：

1. 求得

$$P=\begin{pmatrix}0&0&0&1&0&0\\0&1&0&0&0&0\\0&0&0&0&0&1\\0&0&1&0&0&0\\0&0&0&0&1&0\\1&0&0&0&0&0\end{pmatrix}\quad 与 \quad Q=\begin{pmatrix}0&0&0&0&0&1\\0&1&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&0\\0&0&1&0&0&0\end{pmatrix},$$

使

$$
PBQ = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.
$$

2. $r = (-1, -1, -2, 1, -11)$.

3. $k = 4$, $N_{\cdot 4}$ 进入基.

4. $d_B = (0,0,0,1,-1,0)^T$,

   $\theta = x_{B_5} = 20$, $l = 5$, $B_{\cdot 5}$ 退出基.

5. $B = \{(1,3),(2,4),(4,7),(3,5),(5,6),(4,6)\}$,

   $N = \{(1,2),(1,4),(2,7),(3,6),(6,7)\}$,

   $x_B = (50,70,50,50,20,20)^T$. 如图 9.5.

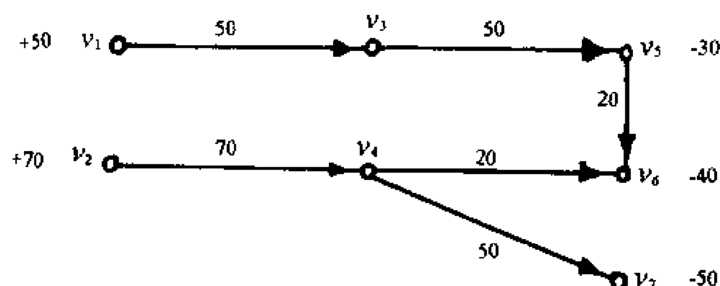


图 9.5

第 3 次迭代:

1. 求得

$$
P = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad 与 \quad Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix},
$$

使

$$
PBQ = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.
$$

2. $r = (-2, -2, -2, -1, -1) \leqslant 0$,

   $\therefore$ 当前解为最优解,即得最优调运方案为

$$x_{13}=50,\ x_{24}=70,\ x_{35}=30,\ x_{46}=20,$$
$$x_{47}=50,\ x_{56}=20,\ 其余\ x_{ij}=0.$$

相应最小总运费为 740.

关于初始可行基

算法 9.5 是从一个可行基开始迭代的. 初始可行基可用人造变量法求出.

今设想把所有供应量集中在发点 1,然后再由发点 1 运送到所有中间点和收点. 即在转运网络图中添加一些边,使

(1) 每个发点 $i(i\neq 1)$ 到发点 1 都有一条边,

(2) 发点 1 到所有中间点和收点都有一条边.

设添加的边的集为 S. $(i,j)\in S$,则称 (i,j) 为**人造边**,称 x_{ij} 为**人造变量**.

由此构成的新的转运网络称为**人造转运网络**. 人造转运网络图 $\overline{G}=(V,E\cup S)$,其基本关联矩阵设为 $\overline{A}$,则人造转运网络规划问题是

$$\begin{aligned}\min&\sum_{(i,j)\in S}x_{ij}\\ \text{s.t. }&\overline{A}x=b,\\ &x\geqslant 0.\end{aligned}\tag{9.58}$$

问题(9.58) 显然有一个可行基 B,

$$B=\{(i,1)\,|\,i\in X\backslash\{1\}\}\cup\{(1,j)\,|\,j\in I\cup Y\},\tag{9.59}$$

其中 X,I 与 Y 分别为发点集,中间点集与收点集,相应基可行解是

$$\begin{cases}x_{i1}=a_i,\ i\in X\backslash\{1\},\\ x_{1j}=\begin{cases}0, & j\in I,\\ b_j, & j\in Y.\end{cases}\end{cases}\tag{9.60}$$

例 9.14 求例 9.13 的转运网络规划的初始可行基.

首先,添加边的集 $S=\{(2,1),(1,5),(1,6),(1,7)\}$. 其次,由一般形式(9.58) 得人造转运网络规划问题:

$$\begin{aligned}\min\ & x_{21}+x_{15}+x_{16}+x_{17}\\ \text{s.t.}\ & x_{12}+x_{13}+x_{14}+x_{15}+x_{16}+x_{17}-x_{21}=50\\ & x_{21}+x_{24}+x_{27}-x_{12}=70\\ & x_{35}+x_{36}-x_{13}=0\\ & x_{46}+x_{47}-x_{14}-x_{24}=0\\ & -x_{15}-x_{35}+x_{56}=-30\\ & -x_{16}-x_{36}-x_{46}-x_{56}+x_{67}=-40,\\ & x_{ij}\geqslant 0.\end{aligned}\tag{9.61}$$

由式(9.59) 与式(9.60),有

$B=\{(2,1),(1,3),(1,4),(1,5),(1,6),(1,7)\}$,

$N=\{(1,2),(2,4),(2,7),(3,5),(3,6),(4,6),(4,7),(5,6),(6,7)\}$,

$x_B=(70,0,0,30,40,50)^T$,如图 9.6.

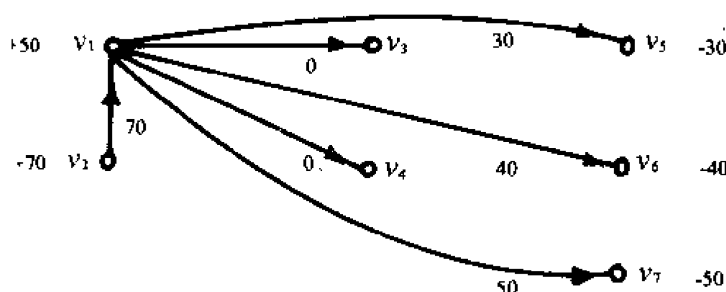


图 9.6

利用算法 9.5，经过 6 次迭代，人造边全部退出基，得转运问题的初始可行基 B 及相应基可行解 x_B：

$$B = \{(4,6),(1,3),(2,4),(3,5),(3,6),(2,7)\},$$

$$x_B = (20,50,20,30,20,50)^T.$$

这正是图 9.3 的生成树及基可行解.

算法 9.6 解转运网络规划的二阶段法

考虑转运网络规划(9.41).

第 1 阶段 解人造转运网络规划.

构造人造转运网络规划(9.58)，以(9.59)的 B 为初始可行基，相应基可行解由式(9.60)给出. 因为人造转运网络规划可行且有界，故必有最优解. 当利用算法 9.5 求得最优解时，

(1) 若最优值为正，则运输网络规划(9.41)不可行；

(2) 若最优值为零，即可得运输网络规划(9.41)的初始基可行解，进入第 2 阶段.

第 2 阶段 解转运网络规划.

由第 1 阶段得到的转运网络规划(9.41)的初始可行基和基可行解，用算法 9.5 求解.

在定理 9.4 中给出了网络规划的重要性质——解的整数性. 读者从解运输网络规划与解分配网络规划的算法可以看到，只需加减运算而没有乘除运算. 对于一般网络规划——转运网络规划，也有同样的性质.

定理 9.24 解网络规划的单纯形法只需加减运算而没有乘除运算(习题 9.53).

下面指出，算法 9.5 和算法 9.6 可以在网络图上进行.

计算基可行解

求解 $Bx_B = b$. 因为 B 对应的边是生成树，生成树必有悬点，故可从生成树的悬点开始考查，利用平衡条件求出悬边对应基变量的值，然后从生成树中去掉该悬点. 继续这一过程可求出所有基变量的值，即得基可行解 x_B.

例 9.15 在例 9.13 中，对转运网络图 9.2，容易找出生成树如图 9.3，试求对应的基可行解.

在图 9.3 中，v_1 是悬点，有供应量 50，故从 v_1 沿悬边 (v_1,v_3) 运送 50 到中间点 v_3，即令 $x_{13} = 50$，然后去掉悬点 v_1，如图 9.7(a). 其余过程如图 9.7 中的(b) ~ (f)，于是得生成树

所决定的基可行解：

$$x_{13}=50,\ x_{35}=30,\ x_{36}=20,\ x_{46}=20,$$
$$x_{24}=20,\ x_{27}=50,\ \text{其余}\ x_{ij}=0.$$

这就是在例 9.13 中的初始基可行解.

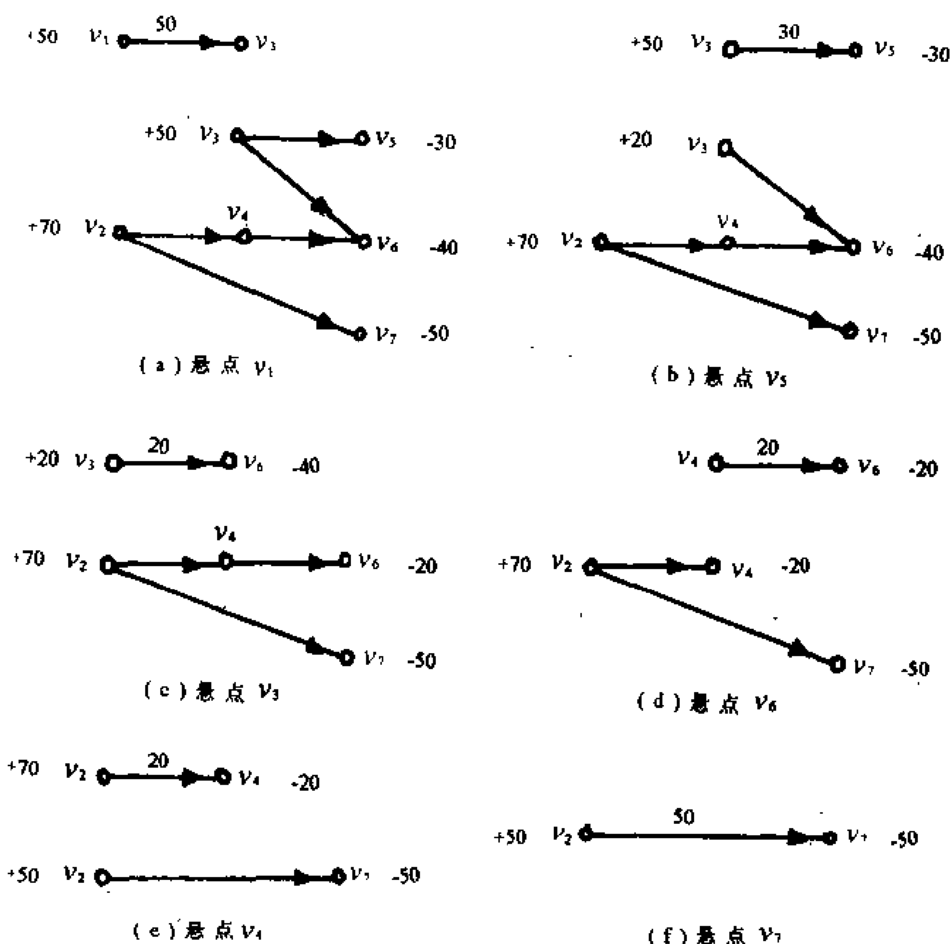


图 9.7　由生成树求基可行解

计算对偶解——顶点的位势

求解 $wB=c_B$. 因为收点 v_n 的平衡条件已去掉，故在生成树中若有边(v_k,v_n)，则 $w_k=c_{kn}$，于是 $w_n=0$. 换言之，令 $w_n=0$，并由

$$w_i-w_j=c_{ij},\ (i,j)\in B \tag{9.62}$$

求出对偶可行解.

例 9.16　在例 9.15 已给出如何利用图 9.3 的生成树求基可行解. 试由此生成树求对偶可行解.

重新画出图 9.3 的生成树，并在有向边(i,j)旁标出权 c_{ij}，如图 9.8.

令 $w_7=0$，由 $w_2-w_7=c_{27}=5$ 得 $w_2=5$. 再由 $w_2-w_4=c_{24}=1$ 得 $w_4=4$. 并相继求出 w_6,w_3,w_5 与 w_1. 计算结果标在图 9.8 顶点旁.

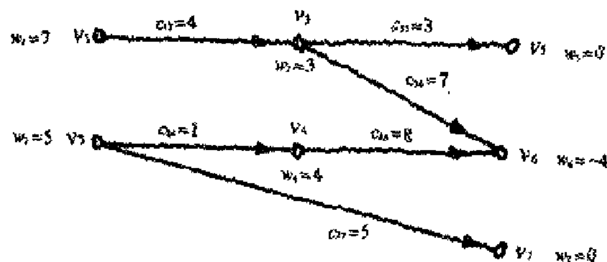


图 9.8　由生成树求对偶可行解

计算判别数

计算 $r = wN - c_N$，即

$$r_{ij} = w_i - w_j - c_{ij},\ (i,j) \in N. \tag{9.63}$$

由于顶点的位势 $w_1,\cdots,w_n$ 已求出，基变量的判别数为零，故由式(9.63)容易求得非基变量的判别数.

决定入基和出基向量

若 $r \leqslant 0$，则当前解为最优解. 否则令

$$r_{kl} = \max\{r_{ij} \mid (i,j) \in N\} > 0, \tag{9.64}$$

得入基向量的边 (k,l). 网络图的生成树加上边 (k,l) 有唯一圈. 设

$$B \cup \{(k,l)\}$$

的唯一圈是 C，C 的方向即边 (k,l) 的方向. 若非基变量 x_{kl} 从零增加到 θ，为保持平衡条件，C 上与 C 方向一致的边都应增加 θ，方向相反的边都应减少 θ. 因此，由非负性，应取 θ 满足条件

$$\theta = x_{rt} = \min\{x_{ij} \mid (i,j) \in C^-\}, \tag{9.65}$$

其中 C^- 是 C 上方向相反的所有边集. 于是出基向量的边是 (r,t).

新的可行基和基可行解

新的可行基是

$$(B \backslash \{(r,t)\}) \cup \{(k,l)\}. \tag{9.66}$$

相应基可行解在圈 C 上调整可得

$$x_{ij} = \begin{cases} x_{ij} + \theta,\ (i,j) \in C^+, \\ x_{ij} - \theta,\ (i,j) \in C^-, \\ x_{ij} \qquad ,\text{其它}, \end{cases} \tag{9.67}$$

其中 C^+，C^- 分别为 C 上方向一致，方向相反的边集.

例 9.17　在例 9.15 中已求出图 9.3 的生成树所决定的基可行解. 试求新的基可行解.

首先由式(9.63)计算非基变量判别数，如图 9.9(a). 再由式(9.64)得入基向量的边为(4,7). 于是 $B \cup \{(4,7)\}$ 有唯一圈 $C = \{(4,7),(2,4),(2,7)\}$，并且 $C^+ = \{(4,7),(2,4)\}$，$C^- = \{(2,7)\}$，如图 9.9(b). 由式(9.65)，出基向量的边是(2,7)，故新的可行基是由(4,7)代替(2,7)，即得

$$\{(1,3),(2,4),(4,7),(3,5),(3,6),(4,6)\}.$$

相应基可行解由式(9.67)得

$$x_{13}=50,\ x_{24}=70,\ x_{47}=50,\ x_{35}=30,$$
$$x_{36}=20,\ x_{46}=20,\ \text{其余}\ x_{ij}=0,$$

如图 9.9(c). 这一结果与例 9.13 的第 1 次迭代完全一致.

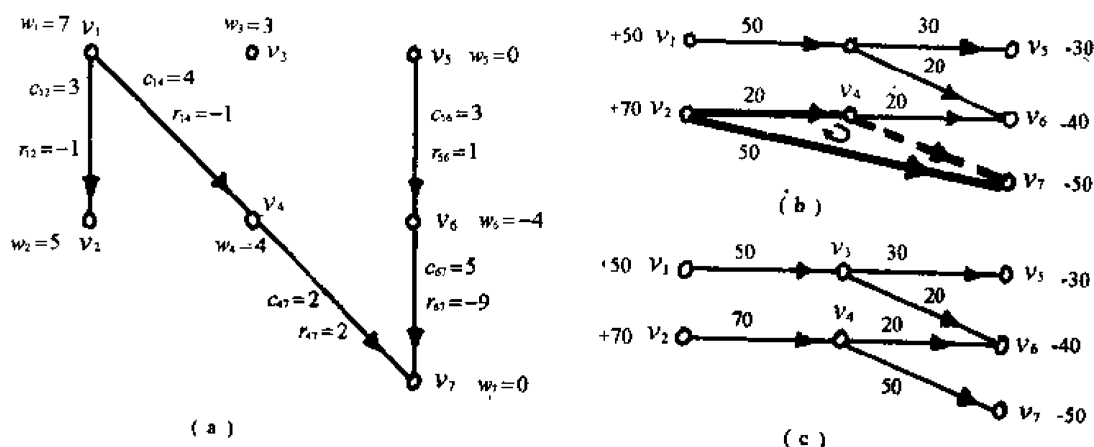


图 9.9 (a) 计算判别数 (b) 圈 C (c) 新基可行解

算法 9.7 转运网络图上的原始-对偶算法

考虑转运网络规划(9.41)及相应网络图 $G=(V,E)$,设 B 是可行基.

第 1 步 求基可行解.

考查生成树 B 的悬点,利用平衡条件求出悬边对应基变量的值,然后去掉该悬点,继续这一过程,直至求出所有基变量的值.

第 2 步 求对偶可行解.

令 $w_n=0$,由式(9.62)求出与顶点 n 相邻的顶点的位势.对刚求出位势的顶点的相邻顶点继续用式(9.62)求出其位势,直至所有顶点的位势被求出.

第 3 步 求判别数.

对连枝对应的非基变量,用式(9.63)求判别数.

第 4 步 最优性检验.

若 $r_{ij}\leqslant 0,(i,j)\in N$,则当前解是最优解.否则由式(9.64)确定入基向量的边 (k,l).

第 5 步 求新的基可行解.

设 $B\cup\{(k,l)\}$ 有唯一圈 C,并以 (k,l) 的方向为 C 的方向.令 C^+,C^- 分别为 C 上方向与 C 一致的边集,方向与 C 相反的边集.由式(9.65)确定出基向量的边 (r,t),新的可行基是(9.66),相应基可行解由式(9.67)计算.

返回第 2 步.

习　题　九

9.1　给定运输表

单价 产地 / 销地	1	2	3	a_i
1	4	8	5	30
2	2	6	3	20
b_j	15	10	25	

试画出网络图并列出网络规划.

9.2　给定分配表

效益 工作 / 人员	1	2	3
1	5	10	7
2	3	5	8
3	9	6	4

试画出网络图并列出网络规划.

9.3　求下列网络从 s 到 t 的最大流，试列出网络规划.

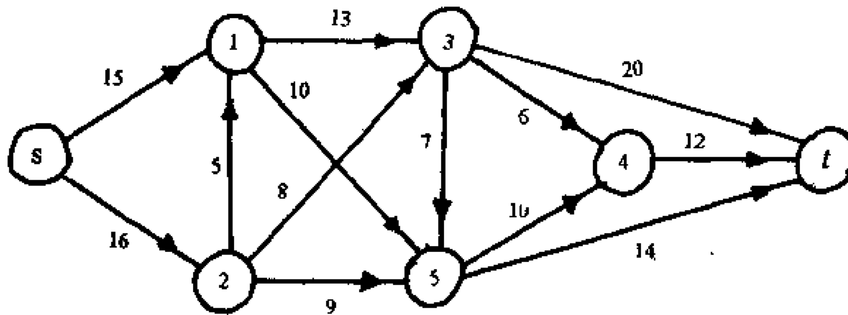


图 9.10

9.4　求下列网络从 v_0 到 v_5 的最短路，试列出网络规划.

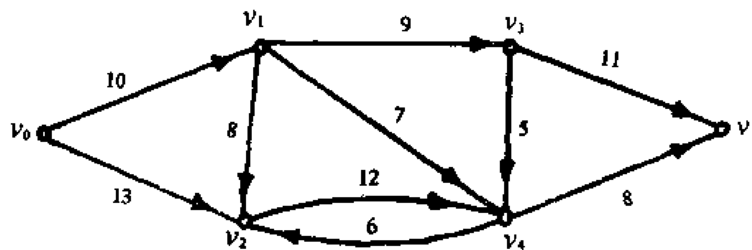


图 9.11

9.5　对下列转运网络,试列出网络规划.

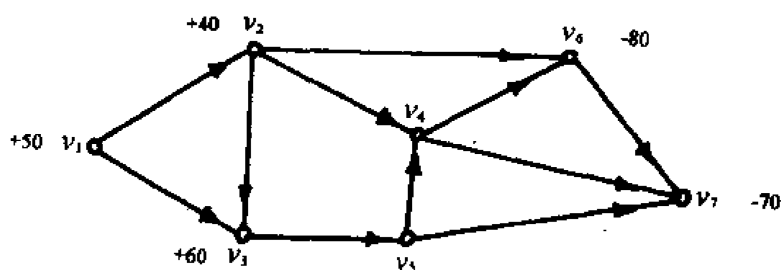


图 9.12

9.6　给定网络图如下,试列出最小费用流网络规划.

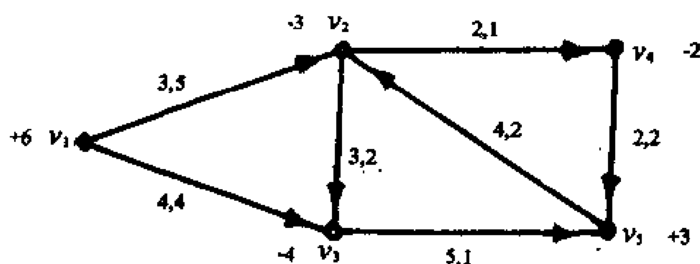


图 9.13
图中箭杆旁的数字为 $a(e),c(e)$

9.7　证明引理 9.1.

9.8　证明引理 9.2.

9.9　设 $A=(a_{ij}),a_{ij}=0$ 或 1 或 -1,若

(1) 每列非零元不多于 2 个.

(2) 行能分解为两个子集 Q_1 与 Q_2,使

1) 若某列含 2 个同号非零元,则分别含于 Q_1 与 Q_2 中;

2) 若某列含 2 个异号非零元,则同时含于 Q_1 或 Q_2 中;

则 A 是全单模的.

9.10　设 $A\in R^{m\times n}$ 是整数的,$r(A)=m$,则下列各条等价.

(1) A 的 m 阶非奇异子矩阵的行列式是 ± 1;

(2) 若 $b\in R^m$ 是整数的,则

$$S\triangleq\{x\mid Ax=b,\ x\geqslant 0\}$$

的基可行解是整数的;

(3) A 的 m 阶非奇异子矩阵的逆是整数的.

9.11　证明引理 9.7.

9.12　证明引理 9.8.

9.13　证明引理 9.9.

9.14　证明引理 9.11.

9.15　证明引理 9.12.

9.16　在运输表中给出点集 $M=\{(1,1),(1,2),(2,1),(3,1),(3,3),(3,4),(3,5),(4,2),(4,3)\}$，试求出 M 的闭回路，并由此推出 P_{42} 的表示式.

9.17　在下面的运输表中给定了一些变量的值，问是否基可行解，为什么？

(1)

6				6
2	10			12
		6	2	8
			4	4
8	10	6	6	

(2)

3			3	6
		7		7
4	1			5
	7		5	12
7	8	7	8	

(3)

			3			5	8
		2		4			6
					3	6	9
6	7						13
	8						8
8			2				10
	4	2					6
14	19	4	5	4	3	11	

9.18　(1) 给出最小元素法的计算步骤，

(2) 用(1)的算法求例 9.8 的初始基可行解.

9.19　给定运输表，分别用西北角法与最小元素法求初始基可行解.

4	3	6	2	14
7	8	5	4	7
6	9	11	10	9
6	8	7	9	

9.20　对运输网络规划，设 P_{kl} 进入基，P_{rt} 退出基，证明新的目标函数值将下降 $x_{rt}r_{kl}$.

9.21　在求单纯形乘子的公式(9.15)中，能否把 $u_1=0$ 改为 $u_i=0$ 或 $v_j=0$？为什么？

9.22　用位势法解下列运输网络规划.

1	2	3	4	6
4	5	2	1	8
2	1	4	3	10
4	5	9	6	

9.23　用位势法求解产大于销的运输网络规划.

4	10	5	10
3	2	8	12
1	9	6	8
9	7	6	

9.24　用位势法求解产小于销的运输网络规划.

2	6	3	9
10	8	5	8
1	5	7	6
10	7	12	

9.25　给定下列产大于销的运输表，并规定产地1的产量必须全部运完，试利用位势法求解.

3	8	7	15
10	6	9	20
10	8	12	

9.26　给定下列产小于销的运输表，并规定销地2必须全部供应，试用位势法求解.

10	8	6	24
7	5	9	12
15	20	10	

9.27 给定产大于销的运输表，并规定多余产量可任意运到某销地，试用位势法求解.

4	6	2	8
3	9	7	7
1	8	10	10
9	6	5	

9.28 已知运输问题

c_{ij} \ j / i	1	2	3	4	5	6	a_i
1	2	1	3	3	3	5	50
2	4	2	2	4	4	4	40
3	3	5	4	2	4	1	60
4	4	2	2	1	2	2	31
b_j	30	50	20	40	30	11	

(1) 求最优运输方案.

(2) 若最优运输方案不变，分别求 c_{13}，c_{24}，c_{35} 与 c_{41} 所满足的条件，假定其余运输单价不变.

(3) 若最优运输方案不变，求 c_{11} 所满足的条件，假定其余运输单价不变.

9.29 甲、乙两煤矿分别生产煤500万吨，供应三个发电厂，A需300万吨，B需300万吨，C需400万吨. 煤可从矿直运厂，也可经由另外的矿或厂转运. 各矿间，厂间，矿与厂间的距离如下表，试用位势法求最优运输方案.

从 \ 到	甲	乙
甲	0	120
乙	100	0

从 \ 到	A	B	C
甲	150	120	80
乙	60	160	40

从 \ 到	A	B	C
A	0	70	100
B	50	0	120
C	100	150	0

9.30 给定运输表，每个销地的销量有2个数，b_j^{min} 表销地 j 必须运进的量，b_j^{max} 为销地 j 的最高需要量，$b_j^{max} \geqslant b_j^{min}$，并假定产地3不能运到销地4，试用位势法求最优运输方案.

	16	13	22	17	50
	14	13	19	15	60
	19	20	23	不通	50
b_j^{min}	30	70	0	10	
b_j^{max}	50	70	30	60	

9.31　给定运输表，每个产地的产量有 2 个数，a_i^{min} 表产地 i 必须运出的量，a_i^{max} 为产地 i 的最大供应量，$a_i^{max} \geqslant a_i^{min}$，并假定产地 3 不能运到销地 4，试用位势法求最优运输方案.

				a_i^{min}	a_i^{max}
16	13	22	17	40	60
14	13	19	15	50	50
19	20	23	不通	50	70
50	60	20	40		

9.32　给定运输表，并假定

(1) 销量刚好被满足，

(2) 产地 1 的产量可以不运出，

(3) 产地 3 的产量必须全部运出，

(4) 产地 2 的产量若不运出则每单位罚 1，

试用位势法求最优运输方案.

2	11	3	4	7
10	3	5	9	5
7	8	1	2	8
2	3	4	6	

9.33　给定运输表，并假定

(1) 产量全部运出，

(2) 销地 1 的需要量必须满足，

(3) 销地 2 的需要量若未满足则每单位罚 2，

(4) 销地 3 的需要量可以不满足，

试用位势法求最优运输方案.

2	5	3	9
10	8	6	8
1	4	7	6
10	7	12	

9.34　给定运输表，并假定

(1) 销地 1 可多接受 2 单位，

(2) 销地 2 与 3 可任意多接受，

试用位势法求最优运输方案.

4	6	2	8
3	9	7	7
1	8	10	10
9	6	5	

9.35　给定运输表如表 9.1，设已求得基最优解 $\{x_{ij} \mid (i,j) \in M\}$，

（1）从某产地 i 运出的单价同时增加或减少一个常数，讨论对最优解与最优值的影响；

（2）设某销地 j 运进的单价同时增加或减少一个常数，讨论对最优解与最优值的影响.

9.36　某厂欲制订机床生产的年度计划，假定每季度末交货，若当季生产的机床未交货，则每台机床每季需支付存储费 300 元. 已知生产数据如下表，试用运输网络规划制订总费用最少的年度计划.

季度	1	2	3	4
交货台数	10	15	25	20
最大生产台数	25	35	30	10
每台生产费用	10800	11100	11000	11300

9.37　用位势法求解下列分配网络规划.

效率＼工作 人员	1	2	3
1	2	5	7
2	4	2	1
3	2	6	5

9.38　游泳接力赛由蛙泳、蝶泳、仰泳与自由泳 4 段组成，每个队员只能参加一段. 已知 4 个队员的各项成绩（时间：秒）如下表. 问如何分配队员能取得最好成绩，即 4 段的总时间最少（假定队员交接时间为零）？试用位势法求解.

时间＼项目 队员	蝶泳	蛙泳	仰泳	自由泳
A	30	31	33	28
B	35	29	36	32
C	32	32	35	33
D	30	33	31	28

9.39　证明引理 9.13.

9.40　证明引理 9.16.

9.41　证明引理 9.17.

9.42　证明引理 9.18.

9.43　证明引理 9.19.

9.44　用匈牙利方法求解分配网络规划：

$$\begin{bmatrix} 4 & 3 & 5 & 1 & 4 \\ 1 & 1 & 3 & 6 & 2 \\ 1 & 5 & 4 & 6 & 1 \\ 3 & 3 & 4 & 1 & 4 \\ 9 & 1 & 2 & 1 & 2 \end{bmatrix}.$$

9.45　用匈牙利方法重解习题 9.37.

9.46　用匈牙利方法重解习题 9.38.

9.47　证明引理 9.20.

9.48　证明引理 9.21.

9.49　证明引理 9.22.

9.50　证明引理 9.23.

9.51　如习题 9.19 的运输表，试用算法 9.6 求解.

9.52　给定转运网络图 9.14，试用二阶段法求解.

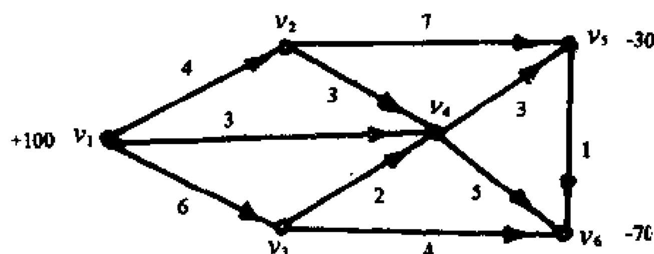


图 9.14

9.53　证明定理 9.24.

9.54　用算法 9.7 解习题 9.52 的转运网络问题.

9.55　给定转运网络图 9.15，引入人造边，用算法 9.7 求解.

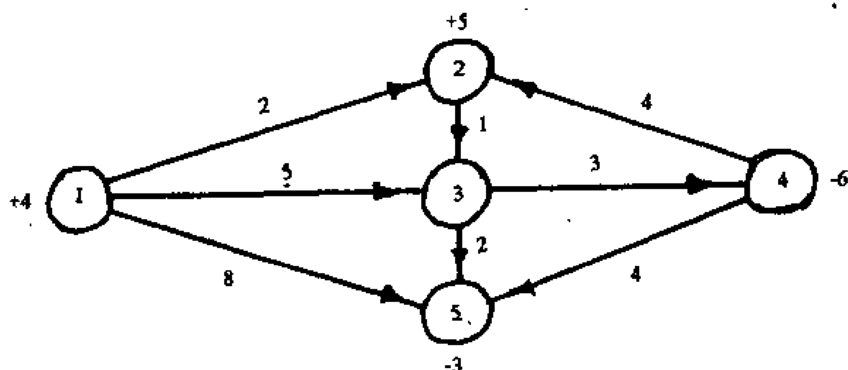


图 9.15

9.56　最小费用流网络规划(9.10)就是转运网络规划再附加变量有上界. 试修改算法 9.7 以解最小费用流网络规划如图 9.16,并求解下列最小费用流问题.

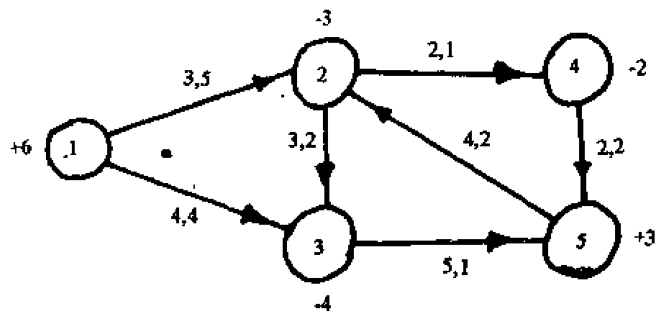


图 9.16
图中箭杆旁的数字为上界 u_{ij},单价 c_{ij}

9.57　在最小费用流网络规划(9.10)中把上界条件 $0 \leqslant x \leqslant u$ 改为上、下界

$$l \leqslant x \leqslant u,$$

如何利用算法 9.7 求解?试求解下列有上、下界的问题如图 9.17.

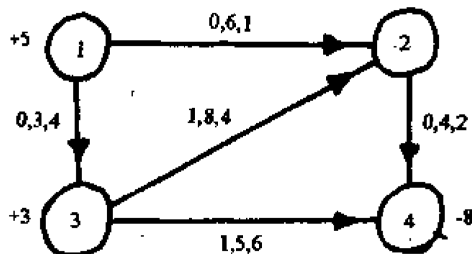


图 9.17
图中箭杆旁数字为下界 l_{ij},上界 u_{ij},单价 c_{ij}

参 考 文 献

[1] Bondy, J. A., and U. S. R. Murty, Graph Theory with Applications, New York, Academic Press, 1976.

[2] Berge, C., Graphs and Hypergraphs, Amsterdam, North-Holland, 1976.

[3] Berge, C., Graphs, Amsterdam, North-Holland, 1985.

[4] Gondran, M., and M. Minoux, Graphs and Algorithms, John Wiley & Sons, 1984.

[5] 陈树柏,网络图论及其应用,北京,科学出版社,1984.

[6] 陈子岐,朱必文,刘峙山,图论,北京,高等教育出版社,1991.

[7] 卢开澄,图论及其应用,北京,清华大学出版社,1981.

[8] Harary, F. (ed.), A Seminar on Graph Theory, Holt, Rinehart and Winston, New Your, 1967.

[9] Harary, F., Graph Theory, Addison-Wesley, Reading, Mass., 1969.

[10] Deo, N., Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1974.

[11] B. Andrásfai 著,郭照人译,图论导引,北京,高等教育出版社,1985.

[12] C. H. Papadimitriou and K. Steiglitz 著,刘振宏,蔡茂诚译,组合最优化,算法和复杂性,北京,清华大学出版社,1988.

[13] 费培之,线性和非线性规划引论及其应用,成都,四川大学出版社,1991.

[14] Bazaraa, M. S. and J. J. Jarvis, Linear Programming and Network Flows, John Wiley & Sons, Inc., 1977.

[15] Biggs, N. L., E. K. Lloyd, and R. J. Wilson, Graph Theory 1736～1936, Oxford University Press, 1976.

[16] Brown, W. G., (ed.), Reviews in Graph Theory, Vol. Ⅰ～Ⅳ, Amer. Math. Soc., 1980

[17] Knuth, D. E., The Art of Computer Programming, Vol. I, Foundamental Algorithms, Reading, Mass., Addison-Wesley Publishing Company, Inc., 1973.

[18] Smith, D. K., Network Optimisation Practice: A Computational Guide, Ellis Horwood Limited, 1982.

[19] Euler, L., Solutio Problemutis ad Geometriam Situs Pertinantis, Academinae Petropolitanae, 8, pp. 128～140, 1736.

[20] Kirchhoff, G. R., Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der Linearen Vertheilung Galvanischer Ströme Geführt wird, Ann alen der Physik und Chemie, Vol. 72, pp. 497～508, 1847.

[21] Cayley, A., On the Theory of the Analytical Forms Called Trees, Philosophical Magazine, (4) 13, pp. 172~176, 1857.

[22] Hamilton, W. R., Account of the Icosian Calcutus, Proc, Roy, Irish. Acad., pp. 415 ~416, 6, 1858.

[23] König, D., Theorie der Endliche und Unendliche Graphen, Leipzig, Akademische Verlagsgesellshaft, 1936.

[24] Wilson, R. J., and .. W. Beinehe (ed.), Applications of Graph Theory, New York, Academic Press, 1979.

[25] Chan, S. P., S. Y. Chan and S. G. Chan, Analysis of Linear Networks and Systems, Reading, Mass., Addison-Wesley, Publishing Company, Inc., 1972.

[26] Lawler, E. L., Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.

[27] Capobianco, M., and J. C. Molluzzo, Examples and Counterexamples in Graph Theory, Elsevier North-Holland, Inc., 1978.

[28] Bollobas, B., Graph Theory, An Introductory Course, Springer-Verlag, New York Inc., 1979.

[29] Bollobas, B., Extremal Graph Theory, Academic Press Inc., 1978.

[30] Bowie, W. S., Applications of Graph Theory in Computer Systems, Int. J. Comp. Info. Soc., Vol. 5, No. 1, pp. 9~31, 1976.

[31] Carré, B., Graphs and Networks, Oxford, Charendon Press, 1979.

[32] Chachra, V., P. M. Ghare and J. M. Moore, Applications of Graph Theory Algorithms, Elsevier, North-Holland, 1979.

[33] Chen, W. K., Applied Graph Theory, Amsterdam, North-Holland, 1976.

[34] Aho, A. V., J. Hoperoft and J. D. Ullman, The Design and Analysis of Computer, Algorithms, Reading, Mass., Addison-Wesley Publishing Company, Inc., 1974.

[35] Tarjan, R. E., Depth-first Search and Linear Graph Algorithms, SIAM. J. Computer, Vol. 1, No. 2, pp. 146~160, 1972.

[36] Minty, G. J., A Simple Algorithm for Listing All the Trees of a Graph, IEEE Trans. on Circuit Theory, Vol. CT-12, No. 1, p120, Mar. 1965.

[37] Paton, K., An Algorithm for Finding a Foundamental Set of Cycles of a Graph, ACM., Comm., Vol. 12, No. 9, pp. 514~518, Sept. 1969.

[38] Paton, K., An Algorithm for the Blocks and Cutnodes of a Graph, ACM., Comm., Vol. 14, No. 7, pp. 468~475, July 1971.

[39] Hopcroft, J., and R. E. Tarjan, Efficient Planarity Testing, ACM., J., Vol. 12, No. 4, pp. 549~568, Oct. 1974.

[40] Auslander, L., and S. V. Parter, On Imbedding Graphs in the Plane, J. Math. & Mech., Vol. 10, No. 3, May, 1961

[41] Hopcroft, J., and R. E. Tarjar, Planarity Testing in Vlog V Steps, Proc. IEIP Congress,

Ljubljama, Booklet Ta-2, pp. 18～23, Aug. 1971.

[42] Even, S., Graph Algorithms, Maryland Computer Science Press, 1979.

[43] Chan, S. P. (ed.), Network Topology and Its Engineering Applications, Taiwan University Press, 1975.

[44] Owens, R. C. and S. P. Chan, Graph Theory 1981-A Surrey of Recent Applications, Conf. Record 15th, Asilomar Conf. on Circuit, Systems and Computers, Pacific Grove, California, 1981.

[45] Swamy, M. N. S., and K. Thulasiraman, Graphs, Networds and Algorithms, New York, John-Wiley and Sons, Inc., 1981.

[46] Tomescu, I., Problems in Combinatorics and Graph Theory, John Wiley & Sons, 1985.

[47] Kuratowski, C., Sur le Probleme des Courbes Gauches en Topologie, Fund. Math. Vol. 15, pp. 271～283, 1930.

[48] Tutte, W. T., How to Draw a Graph, Proc. London Math. Soc. 13, pp. 743～767, 1963.

[49] Dirac, G. A., and S. Schuster, A Theorem of Kuratowski, Nederl. Akad. Wetensch. Proc. Ser. A 57, pp. 343～348, 1954.

[50] Whitney, H., Non-separable and Planar Graphs, Trans. Amer. Math. Soc. 34, pp. 339～362, 1932.

[51] Whitney, H., Planar Graphs, Fund. Math. 21, pp. 73～84, 1933.

[52] Wagner, K., Über eine Eigenschaft der Ebenen Komplexe, Math. Ann. 114, pp. 570～590, 1937.

[53] Harary, F., and W. T. Tutte, A Dual Form of Kuratowski's Theorem, Canad. Math. Bull., 8, pp. 17～20, 1965.

[54] Maclane, S., A Structural Characterization of Planar Combinatorial Graphs, Duke Math. J. 3, pp. 340～472, 1937.

[55] Dunn, W. R. Jr., and S. P. Chan, An Algorithm for Testing the Planarity of a Graph, IEEE Trans. on Circuit Theory, Vol. CT-15, No. 2, pp166～168, June 1968.

[56] Reingold, E. M., J. Nievergelt, and N. Deo, Combinatorial Algorithms: Theory and Practice, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1977.

[57] Cook, S. A., The Complexity of Theorem Proving Procedure, Proc, 3rd ACM. Symp. on Theory of Computing, pp. 151～158, New York, 1971

[58] Karp, R. M., Reducibility among Combinatorial Problems, in Complexity of Computer Computations (R. E. Miller, J. W. Thatcher ed.), Plenum Press, New York, pp. 85～103, 1972.

[59] J. E. Hoperoft, J. D. Ullman 著,莫绍揆,段祥,顾秀芬译,形式语言及其与自动机的关系,科学出版社,1979.

[60] 王浩,数理逻辑通俗讲话,科学出版社,1980.